

5

Abordagem bi-objetivo para o PAA

Como discutido anteriormente, o problema de atribuição de árbitros pode envolver a consideração de diversos objetivos e restrições não contempladas na formulação do PAA básico, dependendo do contexto em que está inserido. No capítulo anterior, discutiram-se outras restrições e funções objetivo que podem ser relevantes para o processo de atribuição de árbitros.

Baseado na natureza multi-objetivo do problema de atribuição de árbitros, este capítulo propõe a formulação da variante bi-PAA, que considera simultaneamente as duas funções objetivo utilizadas nos capítulos anteriores. Em seguida, apresenta um estudo sobre algumas técnicas de otimização multi-objetivo que possam ser aplicadas ao bi-PAA, propondo abordagens exata e aproximadas para a sua resolução. Finalmente, exhibe os resultados computacionais da aplicação destas abordagens na resolução de algumas instâncias teste.

O problema bi-objetivo de atribuição de árbitros (bi-PAA) consiste em atribuir árbitros a todas as posições de arbitragem associadas a partidas agendadas para um dado intervalo de tempo (tipicamente um dia ou um final de semana), minimizando-se (i) a soma sobre todos os árbitros do valor absoluto das diferenças entre o número desejado de partidas e o número efetivamente arbitrado e (ii) a soma sobre todos os árbitros dos tempos de espera entre suas partidas consecutivas, sujeito às restrições (a) a (f) descritas na Seção 2.3.

A próxima seção discute alguns trabalhos encontrados na literatura de otimização multi-objetivo. A Seção 5.2 propõe abordagens de resolução aproximadas para o bi-PAA. Em seguida, na Seção 5.3, são apresentados os resultados computacionais da aplicação das abordagens propostas à resolução de instâncias teste. Finalmente, a Seção 5.4 traz alguns comentários finais sobre o capítulo.

5.1

Otimização multi-objetivo

Técnicas de otimização combinatória têm sido utilizadas com sucesso na resolução de diversas aplicações das mais variadas áreas. Contudo, frequentemente é negligenciado o fato de que aplicações reais estão associadas a processos

de tomada de decisões que exigem, em geral, a consideração de diversos objetivos. Problemas combinatórios com múltiplos objetivos são estudados há vários anos (21), entretanto apenas na década de 90, esta área de pesquisa se tornou mais ativa com um número maior de publicações, como mostra o trabalho de Ehrgott e Gandibleux (17). O restante desta seção apresenta alguns conceitos preliminares sobre otimização multi-objetivo e, em seguida, discute alguns trabalhos encontrados na literatura que propõem métodos exatos e aproximados para a resolução de problemas multi-objetivo.

5.1.1

Noções preliminares

Um problema de otimização multi-objetivo pode ser definido por:

$$\min_{x \in X} z(x) = (z^1(x), \dots, z^p(x))$$

onde $X \subset \mathbb{R}^n$ é o conjunto de soluções viáveis e $z : \mathbb{R}^n \rightarrow \mathbb{R}^p$ é a função objetivo, que representa a imagem do conjunto viável sobre o espaço dos objetivos, dado por $Z = \{z(x) \in \mathbb{R}^p | x \in X\}$. O vetor objetivo $z(x) = (z^1(x), \dots, z^p(x))$ é associado à solução x . Um *ponto ideal* (21) do espaço dos objetivos é definido por $\bar{z} = (\bar{z}^1, \dots, \bar{z}^p)$, com $\bar{z}^q = z^q(\bar{x}^q) = \min\{z^q(x) | x \in X\}$, para $q = 1, \dots, p$, isto é, $\bar{x}^q$ é uma solução ótima para o objetivo q . Em outras palavras, $\bar{z}$ representa o ponto do espaço dos objetivos cujas componentes são os ótimos (mínimos no caso de um problema de minimização) para cada objetivo. Se o ponto ideal corresponde a uma solução viável, então esta solução é ótima para todos os objetivos e é denominada *solução ideal*.

Uma vez que não é comum a existência de soluções viáveis ideais, substitui-se o conceito de otimalidade pelo de *eficiência* ou *Pareto otimalidade*. Uma solução $x \in X$ é dita eficiente (ou Pareto ótima) se não existe uma outra solução $x' \in X$ tal que $z^q(x') \leq z^q(x)$, para $q = 1, \dots, p$, com a desigualdade verificada de forma estrita para pelo menos um dos objetivos. Ou seja, se x é eficiente, nenhuma outra solução é melhor do que x para pelo menos um objetivo e tão boa quanto x para os demais. O vetor objetivo $z(x)$ associado a uma solução eficiente é um ponto *não dominado* no espaço dos objetivos (Z). Resolver um problema combinatório multi-objetivo significa encontrar o conjunto completo de suas soluções eficientes.

Seja E o conjunto das soluções eficientes e ND o conjunto dos pontos não dominados, também chamado de *fronteira não dominada* ou *fronteira de Pareto*. Por abuso de linguagem, diz-se de que uma solução domina outra se o ponto no espaço dos objetivos associado à primeira domina o ponto

relativo à segunda. A Figura 5.1 exibe um exemplo de fronteira de Pareto para um problema bi-objetivo. No exemplo, C representa o ponto ideal, pois os dois objetivos assumem o valor mínimo. No entanto, este ponto não está associado a uma solução viável alguma, representando portanto apenas um ponto de referência. Os pontos A e B estão associados também ao valor mínimo para o objetivo z^1 . Entretanto, como $z^2(B) < z^2(A)$, a solução associada ao ponto B é eficiente, mas a associada ao ponto A não o é. Assim, o conjunto dos pontos não dominados, associados a soluções viáveis para o problema em questão, é dado por $\{B, D, E, F, G, H\}$. As linhas tracejadas representam a borda da envoltória convexa dos pontos não dominados. Considerando-se o conjunto dos pontos não dominados, ordenado não decrescentemente pelo valor de z^1 , diz-se que dois pontos x_1 e x_2 são vizinhos se $z^1(x_1) < z^1(x_2)$ e não existe um ponto x_3 tal que $z^1(x_1) < z^1(x_3) < z^1(x_2)$. No exemplo, os pares de pontos vizinhos são (B, D) , (D, E) , (E, F) , (F, G) e (G, H) . Duas soluções eficientes são ditas vizinhas na fronteira caso os pontos associados a estas soluções no espaço dos objetivos sejam vizinhos.

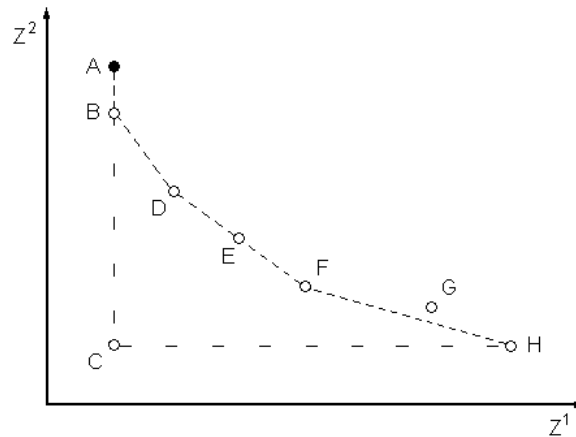


Figura 5.1: Exemplo de fronteira de Pareto.

Diversas soluções eficientes podem corresponder ao mesmo ponto não dominado no espaço dos objetivos. Estas soluções são ditas equivalentes. O número de soluções equivalentes pode ser significativo e sua enumeração intratável, como no caso do problema do caminho mínimo bi-objetivo (38). Considerando-se E como conjunto completo de soluções eficientes, Hansen (38) introduziu a definição do conjunto mínimo completo E_m de soluções eficientes, que consiste em um sub-conjunto de E de cardinalidade mínima que não contém soluções equivalentes e tal que, para toda solução $x \in E$ existe $x' \in E_m$ tal que x e x' são equivalentes. Assim, abordagens de resolução propostas para problemas multi-objetivo visam, em geral, encontrar o conjunto mínimo das soluções eficientes.

A técnica mais utilizada para encontrar soluções eficientes, aplicada em praticamente todos os métodos exatos e em muitas heurísticas, é a escalarização (22). Esta consiste na transformação do problema multi-objetivo em uma sequência de problemas mono-objetivo que são resolvidos repetidamente. Neste processo, diversas técnicas podem ser utilizadas para transformar o vetor de objetivos em uma função objetivo escalar. Em (22, 39), são apresentados estudos sobre algumas técnicas de escalarização de funções para utilização na otimização multi-objetivo.

Os maiores desafios na resolução de problemas multi-objetivo residem no número elevado de soluções eficientes e na existência de soluções eficientes que não são ótimas para escalarização alguma da função objetivo com somas ponderadas. Estas soluções são chamadas *não suportadas* (do inglês *unsupported solutions*), ou seja, soluções cujos pontos correspondentes no espaço objetivo não pertencem à borda da envoltória convexa dos pontos não dominados. Sendo assim, o conjunto de soluções eficientes E pode ser particionado em dois sub-conjuntos ES e EN correspondendo, respectivamente às soluções eficientes suportadas e às não suportadas. No exemplo da Figura 5.1, apenas o ponto G não se encontra sobre a borda da envoltória convexa, de forma que corresponde a uma solução não suportada. Assim como definido para o conjunto E , os conjuntos ES_m e EN_m são os conjuntos mínimos completos das soluções suportadas e não suportadas, respectivamente.

5.1.2

Resolução de problemas combinatórios multi-objetivo

Diversos métodos exatos e aproximados já foram propostos para resolução de problemas combinatórios com múltiplos objetivos. Para revisões da literatura sobre o tema, ver (17, 18, 19, 20, 21, 31, 47, 73).

Os métodos de resolução de problemas multi-objetivo podem ser classificados como *iterativos* ou *não iterativos*. Métodos iterativos alternam fases de computação com fases de comunicação com o usuário, que expressa sua preferência em cada fase de interação e o processo de resolução é direcionado de acordo com esta preferência. Nos métodos não iterativos, busca-se por uma fronteira de soluções eficientes (ou potencialmente eficientes no caso de métodos aproximados), que somente a posteriori será analisada pelo usuário que escolherá a solução (ou um conjunto de soluções) que melhor lhe convier. Nesta tese, apenas métodos não iterativos são propostos e discutidos.

O método mais utilizado para resolução de problemas multi-objetivo é o das somas ponderadas, uma forma de escalarização que minimiza uma

combinação convexa dos p objetivos (22):

$$\min \left\{ \sum_{q=1}^p \lambda_q z^q(x) : x \in X \right\},$$

onde $0 \leq \lambda_q \leq 1$ e $\sum_{q=1}^p \lambda_q = 1$. Variando-se adequadamente os pesos λ_q , $q = 1, \dots, p$, demonstra-se que todas as soluções suportadas podem ser encontradas (35, 44). Apesar de muito utilizado, este método não é capaz de encontrar as soluções não suportadas. Vários trabalhos sobre este método são apresentados em (17).

Uma maneira simples para computar o conjunto mínimo completo de soluções eficientes é dada pelo denominado método dicotômico. No caso de um problema bi-objetivo, o procedimento se inicia pela resolução de dois problemas. O primeiro é a minimização lexicográfica dos objetivos (z^1, z^2) , seguido pela minimização lexicográfica dos objetivos (z^2, z^1) . Minimizar lexicograficamente os objetivos (z^1, z^2) significa encontrar a solução com menor valor para z^1 utilizando-se o valor de z^2 como forma de desempate, ou seja, no caso de soluções com valores iguais para z^1 seleciona-se a solução com menor valor para z^2 . Obtêm-se assim as soluções eficientes que se encontram em cada uma das extremidades da fronteira de Pareto. Seja x^1 a solução eficiente com menor valor para z^1 e x^2 a solução eficiente com menor valor para z^2 . A partir destas duas soluções iniciais, o procedimento resolve recursivamente um novo problema com duas restrições adicionais:

$$z^1(x) \leq z^1(x^1) - \epsilon_1 \quad (5-1)$$

e

$$z^2(x) \leq z^2(x^2) - \epsilon_2, \quad (5-2)$$

que eliminam do conjunto viável as soluções das extremidades da fronteira. Os valores ϵ_1 e ϵ_2 são números reais suficientemente pequenos e são utilizados para transformar a desigualdade estrita em não estrita. A função objetivo do novo problema é então

$$\min \lambda_1 z^1(x) + \lambda_2 z^2(x), \quad (5-3)$$

onde $\lambda_1 = z^2(x^1) - z^2(x^2)$ e $\lambda_2 = z^1(x^2) - z^1(x^1)$.

Os Pseudo-códigos 7 e 8 exibem os passos do método dicotômico, que fornece como resultado o conjunto mínimo completo das soluções eficientes E_m . O Pseudo-código 7 traz o procedimento principal.

O Pseudo-código 8 lista os passos do procedimento que resolve recursivamente o seguinte problema: dadas duas soluções eficientes x^1 e x^2 ($z^1(x^1) < z^1(x^2)$) encontrar uma solução eficiente x^3 tal que $z^1(x^1) < z^1(x^3) < z^1(x^2)$ e de forma a minimizar a função objetivo 5-3. A recursão termina quando

```

1 Pseudo-código MetodoDicotomico()
2 Seja  $x^1$  a solução eficiente resultante da minimização lexicográfica de
   $(z^1, z^2)$ ;
3 Seja  $x^2$  a solução eficiente resultante da minimização lexicográfica de
   $(z^2, z^1)$ ;
4  $E_m \leftarrow \{x^1, x^2\}$ ;
5 ResolvePontoMedioRekursivamente( $x^1, x^2, E_m$ );
6 return  $E_m$ ;

```

Pseudo-código 7: Método dicotômico para resolução exata de problemas multi-objetivo.

nenhuma solução viável existe entre os pontos dados.

```

1 Pseudo-código ResolvePontoMedioRekursivamente( $x^1, x^2, E_m$ )
2 Seja  $x^3$  a solução resultante da resolução do problema de
  programação linear inteira com as restrições adicionais 5-1 e 5-2 e
  que minimiza a função objetivo 5-3;
3 if existe?  $x^3$  then
4    $E_m \leftarrow E_m \cup \{x^3\}$ ;
5    $E_m \leftarrow$  ResolvePontoMedioRekursivamente( $x^1, x^3, E_m$ );
6    $E_m \leftarrow$  ResolvePontoMedioRekursivamente( $x^3, x^2, E_m$ );
7 return  $E_m$ ;

```

Pseudo-código 8: Parte recursiva do método dicotômico.

Através da aplicação do método dicotômico descrito acima, foi possível computar os conjuntos mínimos completos de soluções eficientes para instâncias com até 60 partidas e 120 árbitros do bi-PAA. A resolução exata na linha 2 do Pseudo-código 8 foi feita pelo resolvedor CPLEX versão 8.0 com o Modelo 3 descrito no Capítulo 2. Os resultados obtidos e as respectivas fronteiras de Pareto são exibidos na Seção 5.3.

Outras técnicas para a resolução exata de problemas multi-objetivo são *compromise programming*, programação por metas e o método de duas fases, entre outros. O primeiro foi proposto por Yu (76) e tem como princípio minimizar a distância para um ponto ideal, geralmente utilizando-se algum tipo de normalização como medida da distância. A escolha apropriada dos pesos para a função objetivo permite encontrar todas as soluções eficientes através da resolução de uma sequência de problemas que otimizam funções objetivo normalizadas. A desvantagem do método é que o problema resolvido para cada direção do espaço dos objetivos é, em geral, mais difícil que a variante mono-objetivo do problema original (31).

Como mencionado em (69), o termo programação por metas foi usado pela primeira vez por Charnes e Copper (8). Neste método, que é visto como uma extensão da programação linear (31), o usuário deve definir, para cada objetivo, um valor alvo e pesos para eventuais desvios para mais ou para menos. A função objetivo a ser minimizada é definida como uma soma ponderada e normalizada destes desvios. Uma revisão dos trabalhos sobre programação por metas pode ser encontrada em (69).

O método de duas fases foi proposto em (70). A fronteira de soluções eficientes suportadas é determinada na primeira fase. As soluções eficientes não suportadas são computadas na segunda fase, com base em informações relativas às soluções suportadas que são utilizadas para reduzir o espaço de busca (57). Este método foi utilizado na resolução de vários problemas multi-objetivo como o da mochila bi-objetivo (72), o de atribuição com dois e três objetivos (56, 57) e o de recobrimento de conjuntos bi-objetivo (55).

Com o crescente interesse em problemas multi-objetivo, métodos aproximados para sua resolução começaram a aparecer na literatura a partir de meados da década de 80. Abordagens heurísticas procuram oferecer boas aproximações $\widehat{E}_m$ das fronteiras de Pareto (31) em tempos de processamento razoáveis. Segundo Ehrgott e Gandibleux (17), as primeiras abordagens baseadas em metaheurísticas para problemas multi-objetivo foram os algoritmos genéticos, *simulated annealing*, busca tabu e, mais recentemente, GRASP. Revisões sobre métodos aproximados para problemas multi-objetivo são apresentadas em (17, 19, 21, 47).

5.2

Heurísticas para o bi-PAA

Com o objetivo de encontrar boas aproximações para fronteiras de Pareto de instâncias de tamanho real do bi-PAA, propõe-se uma abordagem heurística para sua resolução, denominada **Heuristica_biPAA**. Esta heurística consiste em um procedimento iterativo baseado em adaptações da estratégia de três fases proposta para as variantes mono-objetivo do PAA. Propõe-se também um procedimento de reconexão por caminhos que é utilizado como uma pós-otimização aplicada sobre pares de soluções da fronteira resultante do algoritmo **Heuristica_biPAA**.

No processo de adaptação para o bi-PAA, os algoritmos que compõem a heurística **3fases-minTE** foram modificados de forma a minimizar uma soma ponderada dos objetivos considerados nesta tese ($z = \lambda_1 z^1 + \lambda_2 z^2$, onde z^1 representa a soma sobre todos os árbitros da diferença entre o número desejado de partidas e o número de partidas arbitradas e z^2 é a soma sobre todos os

árbitros do tempo total de espera entre suas partidas consecutivas).

Em aplicações multi-objetivo, é comum interpretar um conjunto de pesos de uma soma ponderada de vários objetivos como uma direção de busca no espaço dos objetivos. Neste caso, pode-se dizer que a heurística de três fases proposta para o PAA básico (**3fases**) atua na direção $(1, 0)$ e a heurística para a versão que minimiza os tempos de espera (**3fases-minTE**) atua na direção $(0, 1)$. Portanto, a heurística **3fases-biObj** constitui uma generalização das anteriores, uma vez que a direção de busca utilizada é parametrizada pelo par de pesos (λ_1, λ_2) .

A estratégia proposta pelo algoritmo **Heuristica_biPAA** consiste em aplicar a heurística de três fases para um certo conjunto de direções de busca, armazenando-se todas as soluções potencialmente eficientes encontradas ao longo do processo. A resposta dessa heurística é uma aproximação $\widehat{E}_m$ para a fronteira de Pareto da instância de entrada. Essa heurística pode ser utilizada tanto isoladamente, como em conjunto com o procedimento de reconexão por caminhos proposto.

A próxima seção descreve as adaptações feitas na heurística construtiva e no procedimento de perturbação utilizado na heurística aprimorante da estratégia de três fases. Em seguida, na Seção 5.2.2, apresentam-se mais detalhes sobre a abordagem aproximada proposta. Por fim, a Seção 5.2.3 apresenta uma descrição da heurística de reconexão por caminhos proposta para o bi-PAA.

5.2.1

Heurísticas de três fases para o bi-PAA

O algoritmo **CEO_II**, originalmente proposto para o PAA básico, foi a heurística construtiva escolhida para adaptação para a resolução do bi-PAA, pois obteve bons resultados para as variantes mono-objetivo e por ser flexível quanto à função de custo utilizada. Assim como no caso do PAA-minTE, sua adaptação para a variante bi-objetivo consiste simplesmente na substituição dos custos associados às possíveis sequências de PA's para cada árbitro pela soma ponderada dos objetivos. Esta heurística será referenciada nos experimentos por **CEO-biObj**.

Observa-se que a fase de reparação não necessita de alterações, uma vez que visa apenas a viabilidade. Com relação à fase de melhoria, a estrutura proposta é a mesma da heurística para o PAA básico, com alterações somente no procedimento de perturbação, concebido com base nos procedimentos propostos para as variantes mono-objetivo. O Pseudo-código 9 exhibe os passos deste procedimento, cuja idéia é bastante simples e consiste em escolher e

executar, a cada iteração, um dos procedimentos propostos para as variantes mono-objetivo. O critério utilizado na escolha do procedimento é baseado nos pesos dados a cada objetivo e no número de vezes que cada procedimento já foi escolhido até o momento. O intuito deste critério é que a razão entre o número de vezes que cada procedimento é executado seja igual à razão entre os pesos dados a cada objetivo, ou seja, se o peso associado a z^1 é o dobro do peso associado a z^2 , o procedimento de perturbação da variante que minimiza o primeiro objetivo deve ser executado aproximadamente o dobro do número de vezes que o procedimento de perturbação que minimiza o outro objetivo. Assim como nas heurísticas aprimorantes mono-objetivo, a solução obtida após a perturbação seguida de uma busca local é sempre aceita. No Pseudo-código 9, `IterTotal` representa o número total de iterações já executadas até o momento e `IterMin_TE` o número de vezes em que o procedimento escolhido foi o proveniente da heurística que minimiza os tempos de espera. O teste da linha 2 verifica se o número `IterMin_TE` de vezes que o procedimento `Perturbacao_minTE` foi escolhido é menor que o produto entre o número total de iterações já realizadas (`IterTotal`) e a razão entre o peso λ_2 associado a z^2 e a soma $\lambda_1 + \lambda_2$ dos dois pesos. `Perturbacao_minTE` deve ser escolhido, em caso afirmativo, e `Perturbacao_minDif` em caso contrário.

Outra modificação realizada na heurística aprimorante diz respeito ao armazenamento, em uma estrutura de dados especial descrita adiante, de todas as soluções potencialmente eficientes encontradas no decorrer da otimização.

1 Pseudo-código

```

  Perturbacao_biObj(Solucao,  $\lambda_1$ ,  $\lambda_2$ , IterTotal, IterMin_TE)
2  if IterMin_TE <  $\lambda_2 / (\lambda_1 + \lambda_2) * \text{IterTotal}$  then
3    IterMin_TE  $\leftarrow$  IterMin_TE + 1;
4    Perturbacao_minTE(Solucao);
5  else
6    Perturbacao_minDif(Solucao);
7  end
8  IterTotal  $\leftarrow$  IterTotal + 1;
9  return Solucao;
```

Pseudo-código 9: Procedimento de perturbação para a heurística aprimorante.

O procedimento de busca local executado após cada perturbação na heurística `3fases-biObj` é idêntico ao proposto para a variante `3fases-minTE`, substituindo-se apenas a sua função objetivo. Assim como nos casos mono-objetivo, esse procedimento de busca local baseado em vizinhanças pode ser substituído pela resolução exata de um modelo de programação linear inteira.

No caso bi-objetivo, propõem-se as mesmas variantes daquelas propostas para o PAA-minTE, e aqui denominadas, respectivamente: (1) **3fases-biObj** é a variante que utiliza a busca local tradicional por vizinhanças após cada perturbação; (2) **3fases-biObj + MIP** é a variante que substitui a busca local tradicional pela resolução exata em todas as iterações; e (3) **3fases-biObj + MIP_II** é a variante que utiliza um esquema híbrido combinando busca local e resolução exata com uma frequência menor que a da heurística anterior. Esta última variante segue os mesmos passos na heurística proposta para o PAA-minTE ilustrada no Pseudo-código 6. Todas as variantes utilizam o algoritmo construtivo **CEO-biObj** em todos os experimentos.

5.2.2

Algoritmo Heurística_biPAA

A abordagem heurística proposta para o bi-PAA consiste em executar iterativamente uma das heurísticas de três fases adaptadas para um certo conjunto de direções de busca (pares de pesos para a soma ponderada), armazenando-se ao longo do processo todas as soluções potencialmente eficientes encontradas. Uma estrutura de dados especial, descrita da Seção 5.2.4, é utilizada para armazenar o conjunto de soluções não dominadas que começa vazio e é atualizado durante todo o processo de otimização, formando ao final uma fronteira de soluções potencialmente eficientes. O Pseudo-código 10 exhibe os passos desta abordagem.

O algoritmo recebe como parâmetro o número **MaxDirecoes** de direções, o método de três fases que será utilizado no processo de otimização e o critério para o cálculo dos pesos associados a cada objetivo. O método de três fases deve ser escolhido dentre as opções **3fases-biObj**, **3fases-biObj + MIP** ou **3fases-biObj + MIP_II**, descritos na seção anterior. Já os critérios disponíveis para o cálculo dos pesos são *preenchimento de espaços* e *recursivo* e são descritos adiante.

Nas linhas 2 e 3, executa-se a heurística de três fases com o objetivo de minimizar lexicograficamente respectivamente as combinações dos objetivos (z^1, z^2) e (z^2, z^1) . Os pesos ϵ_1 e ϵ_2 são números suficientemente pequenos para que o objetivo associado a tais pesos seja considerado apenas como um fator de desempate entre soluções com valores iguais para o outro objetivo. Em seguida, o laço das linhas 4 a 7 aplica a heurística de três fases para as **MaxDirecoes** – 2 direções restantes (as direções $(1, \epsilon_1)$ e $(\epsilon_2, 1)$ já foram utilizadas nas linhas 2 e 3). A heurística de três fases é executada na linha 10, considerando-se o par de pesos determinado na linha 6 ou 8, segundo um dentre os dois critérios descritos a seguir. Finalmente, na linha 12, é retornado o conjunto de soluções

```

1 Pseudo-código
  Heuristica_biPAA(MaxDirecoes, métodoDe3fases, critério)
2 métodoDe3fases( $\widehat{E}_m$ , 1,  $\epsilon_1$ );
3 métodoDe3fases( $\widehat{E}_m$ ,  $\epsilon_2$ , 1);
4 for  $i \leftarrow 1$  to MaxDirecoes - 2 do
5   if critério = recursivo then
6     Sejam  $\lambda_1$  e  $\lambda_2$  os pesos calculados segundo o critério recursivo;
7   else
8     Sejam  $\lambda_1$  e  $\lambda_2$  os pesos calculados segundo o critério
      preenchimento de espaços;
9   end
10  métodoDe3fases( $\widehat{E}_m$ ,  $\lambda_1$ ,  $\lambda_2$ );
11 end
12 return  $\widehat{E}_m$ ;

```

Pseudo-código 10: Heuristica_biPAA.

potencialmente eficientes encontradas.

Os critérios para determinação, a cada iteração da heurística, dos pesos associados a cada objetivo são denominados de *preenchimento de espaços* e de *recursivo*. Ambos determinam os valores para cada peso com base nas soluções presentes no conjunto de soluções não dominadas no momento da escolha. Duas soluções x^1 e x^2 vizinhas na fronteira são escolhidas e os pesos são calculados da mesma forma que no método dicotômico, dados por $\lambda_1 = z^2(x^1) - z^2(x^2)$ e $\lambda_2 = z^1(x^2) - z^1(x^1)$, onde $z^1(x^2) > z^1(x^1)$ e $z^2(x^1) > z^2(x^2)$ (a solução x^1 é a que tem menor valor para o primeiro objetivo e a solução x^2 é a que tem menor valor para o segundo objetivo). O que difere de um método para o outro é a escolha das soluções utilizadas no cálculo.

O critério *recursivo* seleciona os pares de soluções da mesma maneira que o método dicotômico. Na primeira execução do laço das linhas 4 a 7 do Pseudo-código 10, selecionam-se as duas soluções x^1 e x^2 extremas da fronteira. Se x^3 é a melhor solução encontrada na direção dada pelos pesos calculados a partir de x^1 e x^2 , as próximas duas execuções terão pesos calculados a partir de (x^1, x^3) e (x^3, x^2) , respectivamente. O algoritmo segue fazendo as escolhas de forma recursiva.

No critério por *preenchimento de espaços*, são calculadas as distâncias entre cada par de soluções vizinhas na fronteira e é escolhido o par que apresenta a maior distância. Caso a aplicação da heurística de três fases com os pesos calculados a partir destas soluções não encontre uma solução entre x^1 e x^2 , elas seriam novamente selecionadas na iteração seguinte. Neste caso, para se evitar que o algoritmo cicle, uma lista com pares proibidos de soluções

é mantida e armazena os dez últimos pares quaisquer de soluções selecionadas.

5.2.3

Reconexão por caminhos

O método de reconexão por caminhos (*path relinking*) foi originalmente proposto por Glover (36) como uma forma de intensificação num algoritmo de busca tabu. Ele consiste em explorar trajetórias no espaço de busca que conectam soluções de elite. Dadas duas soluções de elite, o método determina quais atributos são diferentes entre elas. Uma delas é denominada *solução de partida*, sendo a outra denominada *solução guia*. Iniciando-se pela solução de partida, o método realiza, a cada passo, um movimento que insere na solução atual um atributo presente na solução guia e ausente na solução de partida. Assim, a cada passo, a nova solução atual apresenta mais semelhança com relação à solução guia. Resende e Ribeiro (62) descrevem diversas abordagens de reconexão por caminhos encontradas na literatura.

Este método já foi utilizado na resolução de problemas multi-objetivo (21). Nesta tese, propõe-se uma técnica de pós-otimização que consiste na aplicação de um procedimento de reconexão por caminhos a um certo número de pares de soluções extraídas aleatoriamente da fronteira resultante do algoritmo *Heuristica_biPAA*. Para cada par de soluções selecionado, o procedimento apresentado no Pseudo-código 11 é aplicado duas vezes, cada uma partindo de uma das soluções e tendo a outra como guia. A função objetivo utilizada em cada aplicação considera os mesmos pesos (λ_1 e λ_2) que aqueles utilizados para encontrar a solução de partida associada a execução em questão.

Uma vez definidos os árbitros a atuar em cada localidade, o problema de decidir quais posições de arbitragem serão atribuídas a cada árbitro pode ser resolvido de maneira exata, mesmo para instâncias de grande porte com até 500 partidas e 750 árbitros. Os resultados exibidos pelas heurísticas *3fases_II + CEO_II + MIP* e *3fases-minTE + CEO-minTE + MIP* mostraram, nos capítulos anteriores para as variantes mono-objetivo do PAA, que um resolvidor de programação inteira é capaz de resolver o problema associado a cada localidade separadamente. Assim, o maior desafio na resolução do PAA torna-se decidir em qual localidade cada árbitro deverá atuar. Uma heurística de reconexão por caminhos consiste em incorporar iterativamente atributos da solução guia à solução de partida, de forma a visitar uma sequência de soluções intermediárias presentes em um dos caminhos no espaço de busca que conectam as soluções de partida e guia. Assim, o procedimento de reconexão por caminhos proposto para o bi-PAA considera como um atributo de uma solução, a infor-

```

1  Pseudo-código PathRelinking( $\widehat{E}_m, S^{partida}, S^{guia}, \text{MaxIterBL}$ )
2  Seja  $D$  o conjunto de árbitros que atuam em localidades diferentes
   nas soluções  $S^{partida}$  e  $S^{guia}$ ;
3   $S^{atual} \leftarrow S^{partida}$ ;
4  repeat
5      algumMovimento  $\leftarrow 0$ ;
6      algumMovimentoAprimorante  $\leftarrow 0$ ;
7       $S^{melhor} \leftarrow \emptyset$ ;
8      foreach  $(i, i') : i \in D, i' \in D$  while
        algumMovimentoAprimorante = 0 do
9          Sejam  $f_i^{partida}$  e  $f_i^{guia}$  as localidades onde o árbitro  $i$  atua,
            respectivamente, na solução  $S^{partida}$  e na solução  $S^{guia}$ ;
10         Sejam  $f_{i'}^{partida}$  e  $f_{i'}^{guia}$  as localidades onde o árbitro  $i'$  atua,
            respectivamente, na solução  $S^{partida}$  e na solução  $S^{guia}$ ;
11         if  $(f_i^{guia} = f_{i'}^{partida} \text{ or } f_{i'}^{guia} = f_i^{partida})$  e (é viável a troca de
            atribuições entre os árbitros  $i$  e  $i'$ ) then
12             Seja  $S^{nova}$  a solução obtida pela troca das atribuições entre
                 $i$  e  $i'$ ;
13              $S^{nova} \leftarrow \text{BuscaLocal}(f_i^{partida}, S^{nova}, \text{MaxIterBL})$ ;
14              $S^{nova} \leftarrow \text{BuscaLocal}(f_{i'}^{partida}, S^{nova}, \text{MaxIterBL})$ ;
15             if algumMovimento = 0 or o valor objetivo de  $S^{melhor}$  é
                maior que o de  $S^{nova}$  then
16                  $S^{melhor} \leftarrow S^{nova}$ ;
17                 algumMovimento  $\leftarrow 1$ ;
18                 if o valor objetivo de  $S^{melhor}$  é menor que o de  $S^{atual}$ 
                    then
19                     algumMovimentoAprimorante  $\leftarrow 1$ ;
20             end
21         if algumMovimento  $\neq 0$  then
22              $S^{atual} \leftarrow S^{melhor}$ ;
23             InsereQuadTree( $S^{melhor}, \widehat{E}_m$ );
24             if  $i \in D$  then
25                  $D \leftarrow D - \{i\}$ ;
26             if  $i' \in D$  then
27                  $D \leftarrow D - \{i'\}$ ;
28 until  $D = \emptyset$  or algumMovimento = 0 ;
29 return  $\widehat{E}_m$ ;

```

Pseudo-código 11: Reconexão por caminhos para o bi-PAA.

mação de qual localidade abriga as partidas de um determinado árbitro. Ou seja, considera-se como uma diferença entre duas soluções S_1 e S_2 , o fato de um determinado árbitro estar designado a partidas em uma localidade f_1 na solução S_1 e a outra localidade $f_2 \neq f_1$ na solução S_2 .

Seja D o conjunto dos árbitros que atuam em diferentes localidades nas

soluções de partida e guia, respectivamente S^{partida} e S^{guia} . O objetivo do procedimento apresentado no Pseudo-código 11 é, a cada passo, realizar um movimento que troca todas as atribuições entre dois árbitros pertencentes ao conjunto D de forma que, após o movimento, pelo menos um deles estará designado a arbitrar partidas na mesma localidade a que este árbitro foi designado na solução S^{guia} . Nas linhas 2 e 3, são inicializados respectivamente o conjunto D e a solução atual S^{atual} , que é igual à solução de partida. O laço das linhas 4 a 28 percorre as soluções no caminho entre a solução de partida e a solução guia, enquanto a solução atual contiver atributos diferentes da solução guia ($D \neq \emptyset$) e existir algum movimento viável envolvendo dois árbitros de D . Nas linhas 5 e 6, são inicializadas respectivamente as variáveis `algumMovimento` e `algumMovimentoAprimorante`, indicando que nenhum movimento viável foi encontrado até o momento na iteração atual. Na linha 7, é inicializada a variável S^{melhor} que representa a melhor solução vizinha encontrada até o momento, também na iteração atual.

O laço das linhas 8 a 20, procura pares de árbitros (i, i') pertencentes a D , cuja troca das atribuições seja viável e aprimorante, e de forma que após o movimento pelo menos um dos árbitros seja designado a arbitrar partidas na mesma localidade em que atua na solução S^{guia} . Caso nenhum movimento aprimorante seja encontrado, seleciona-se aquele que aumenta menos o valor objetivo da solução atual. Nas linhas 9 e 10, são determinadas as localidades onde atuam os árbitros i e i' nas soluções de partida e guia. O teste na linha 11 verifica se a troca das atribuições entre estes árbitros é viável e se pelo menos um deles, ao final do movimento, estará designado a arbitrar partidas na mesma localidade em que atua na solução guia. Em caso afirmativo, na linha 12 o movimento de troca é realizado e nas linhas 13 e 14 aplica-se uma busca local sobre a solução resultante, considerando-se separadamente cada localidade envolvida da troca. Se a nova solução S^{nova} é a primeira viável ou é o melhor vizinho encontrado até o momento (teste na linha 15), esta solução é armazenada em S^{melhor} . Na linha 17, atualiza-se o valor de `algumMovimento`, indicando que pelo menos um movimento viável foi encontrado. Na linha 19, a variável `algumMovimentoAprimorante` é atualizada, caso a solução S^{melhor} seja aprimorante em relação à solução atual S^{atual} (teste da linha 18).

O teste da linha 21 verifica se algum movimento viável foi encontrado. Em caso afirmativo, a solução S^{atual} é atualizada na linha 22. Caso S^{melhor} seja não dominada pelas soluções atualmente na fronteira, ela é inserida em $\widehat{E}_m$ na linha 23, mantendo-se a propriedade do conjunto de não conter soluções dominadas. Nas linhas 24 a 27 remove-se do conjunto D os árbitros i e i' caso pertençam a este conjunto. O procedimento retorna a fronteira atualizada de

soluções potencialmente eficientes na linha 29.

Assim como no procedimento de *look ahead* proposto na heurística aprimorante para o PAA básico, constatou-se que a busca exaustiva por um par de árbitros cuja troca seja viável e aprimorante (laço das linhas 8 a 20) é extremamente custosa. Neste caso, decidiu-se por limitar essa busca ao número máximo de 25 pares viáveis ou 2% da vizinhança, o que ocorrer primeiro.

5.2.4

Estrutura de QuadTrees

A estrutura de dados utilizada para armazenar as soluções componentes de uma fronteira de Pareto se baseia em uma estrutura denominada *QuadTree*. Esta estrutura, originalmente proposta por Finkel e Bentley (28), é muito utilizada para trabalhar com funções multidimensionais e já foi utilizada em diversas aplicações, por exemplo em bancos de dados e processamento de imagens (40). Ela permite uma indexação em múltiplas dimensões e seu princípio básico consiste em armazenar os elementos do conjunto em uma estrutura de árvore na qual cada elemento apresenta uma certa relação com o seu elemento *pai* em cada dimensão.

Quadtrees foram aplicadas à otimização multi-objetivo pela primeira vez por Habenicht (37). Nesta tese, assim como em (37), foi utilizada uma estrutura denominada *domination-free quadtree*, ou seja, uma quadtree na qual nenhum de seus elementos domina algum outro (também uma propriedade das soluções nas fronteiras de Pareto). Esta estrutura de dados tem como objetivo fornecer formas eficientes de determinar se uma nova solução é dominada por alguma da fronteira atual e de inserir uma solução não-dominada na estrutura, mantendo-se sua propriedade de não possuir soluções dominadas. O procedimento $\text{InsereQuadTree}(\mathbf{S}, \widehat{E}_m)$, utilizado nos algoritmos propostos neste capítulo, procede verificando se a solução $\mathbf{S}$ é potencialmente eficiente. Em caso afirmativo, $\mathbf{S}$ é inserida em $\widehat{E}_m$ e são removidas desse conjunto todas as soluções dominadas por $\mathbf{S}$.

5.3

Resultados computacionais

Os resultados experimentais relatados nesta seção foram obtidos com as mesmas plataformas descritas nos capítulos anteriores. Os experimentos envolvendo a resolução exata dos modelos de programação inteira foram realizados na plataforma descrita no Capítulo 2 e os demais experimentos na plataforma descrita no Capítulo 3.

Foram selecionadas seis instâncias teste, dentre aquelas que foram descritas no Capítulo 2, para os experimentos apresentados neste capítulo, pois apresentaram resultados bem representativos daqueles observados para o grupo completo de instâncias. Os experimentos que envolveram apenas métodos aproximados foram feitos sobre três instâncias com 500 partidas e 750 árbitros e diferentes números de localidades e padrões para geração do número desejado de partidas de cada árbitro. Para a comparação da abordagem aproximada que obteve os melhores resultados com as fronteiras de Pareto computadas de maneira exata pelo método dicotômico, foram utilizadas três instâncias de médio porte com 50 partidas e 100 árbitros (uma instância) e com 60 partidas e 120 árbitros (duas instâncias com diferentes padrões para geração do número desejado de partidas de cada árbitro).

Em todos os casos, para cada direção de busca considerada, foram realizadas cinco execuções com diferentes sementes para geração de números aleatórios. De forma a manter constante o tempo total gasto com cada instância, independentemente do número de direções, o limite máximo de tempo dado à heurística de três fases, para cada execução de cada direção, foi de 800, 400 e 200 segundos para os experimentos com, respectivamente, 16, 32 e 64 direções para as instâncias com 500 partidas e 750 árbitros. Para a instância com 50 partidas e 100 árbitros, o limite de tempo dado foi de 320 e 160 segundos para os experimentos com, respectivamente, 32 e 64 direções. Finalmente, um limite máximo de tempo de 400 e 200 segundos foi dado para as mesmas direções para as instâncias com 60 partidas e 120 árbitros, respectivamente.

Como mencionado por Ehrgott e Gandibleux (21), diversas idéias são propostas na literatura para avaliar e comparar a qualidade de aproximações de fronteiras de Pareto. Entretanto, nenhuma dessas medidas de qualidade é universalmente aceita. Uma vez que não é objetivo deste trabalho o desenvolvimento ou a investigação de métricas para a avaliação de fronteiras aproximadas, decidiu-se pela comparação visual, que já foi utilizada como forma de avaliação de fronteiras (21).

5.3.1

Crítérios para determinação dos pesos

O primeiro experimento apresentado neste capítulo teve o intuito de realizar uma comparação entre os critérios para determinação, em cada iteração da heurística para o bi-PAA, das soluções a serem utilizadas no cálculo dos pesos associados a cada objetivo. Os critérios avaliados foram os de *preenchimento de espaços* e o *recursivo*, descritos na Seção 5.2.2.

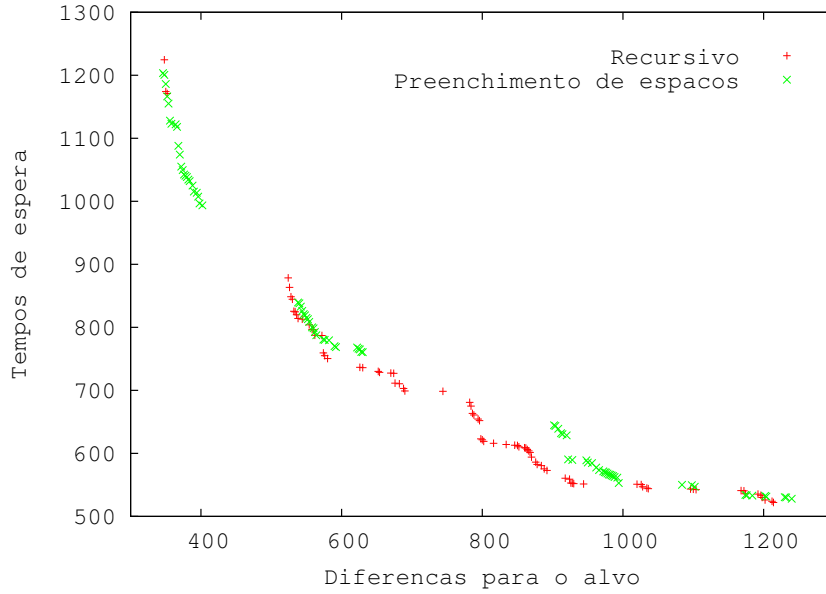


Figura 5.2: Instância com 65 localidades e padrão P_0 , 16 direções de busca.

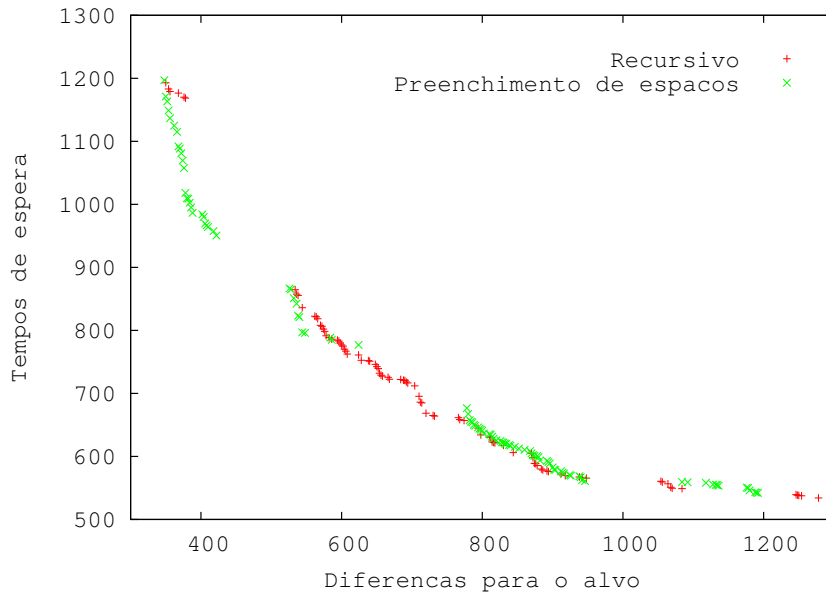


Figura 5.3: Instância com 65 localidades e padrão P_0 , 32 direções de busca.

As Figuras 5.2 a 5.7 exibem as fronteiras de soluções resultantes das resoluções das três instâncias teste com 500 partidas e 750 árbitros. As curvas exibidas nas figuras foram obtidas pelo algoritmo *Heuristica_biPAA* associado à heurística *3fases-biObj* utilizando os critérios de preenchimento de espaços e recursivo. Cada figura exhibe duas curvas, uma para cada critério. Para cada instância, foram utilizadas 16 e 32 direções de busca, resultando nas seis figuras apresentadas.

Observa-se pelas figuras que as fronteiras produzidas pelo método recursivo se apresentam mais homogêneas e com espaços menores (regiões da fron-

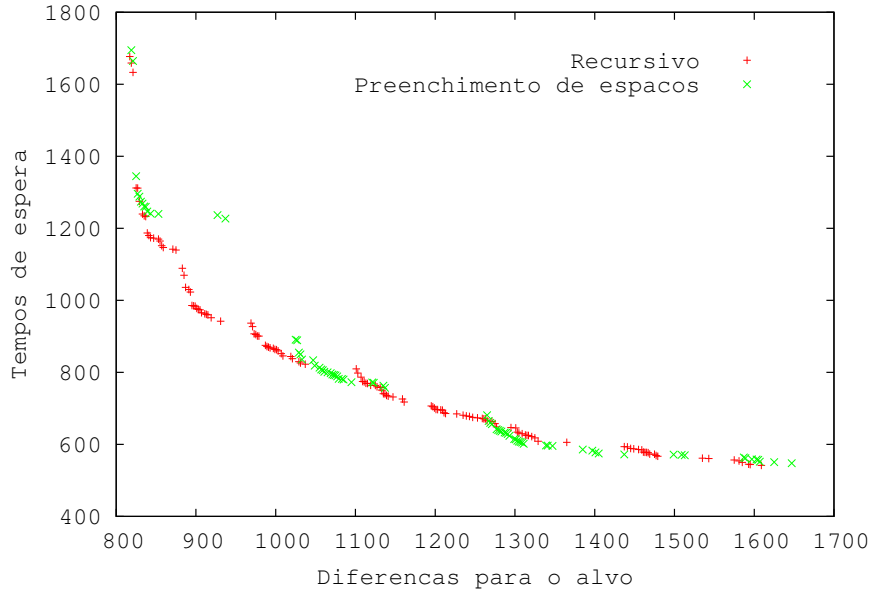


Figura 5.4: Instância com 65 localidades e padrão P_1 , 16 direções de busca.

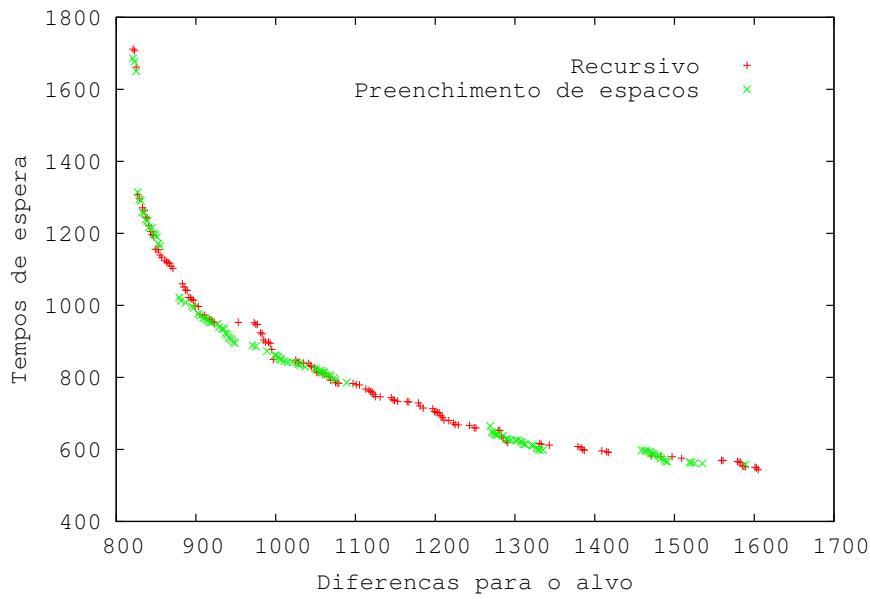


Figura 5.5: Instância com 65 localidades e padrão P_1 , 32 direções de busca.

teira sem soluções) do que as fronteiras obtidas com a utilização do método de preenchimento de espaços. Uma vez que as soluções de uma fronteira não são dominadas pelas da outra, o método adotado para os próximos experimentos foi o *recursivo*.

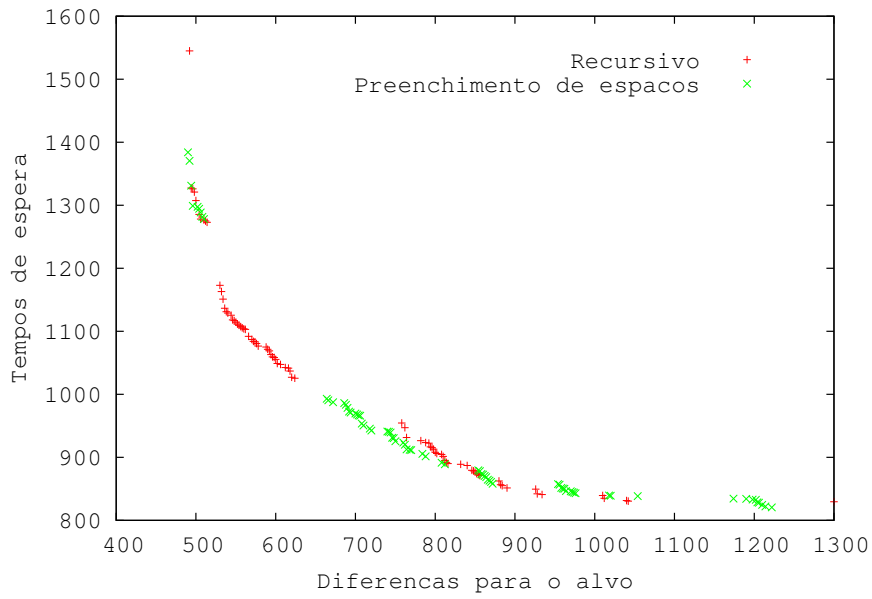


Figura 5.6: Instância com 85 localidades e padrão P_0 , 16 direções de busca.

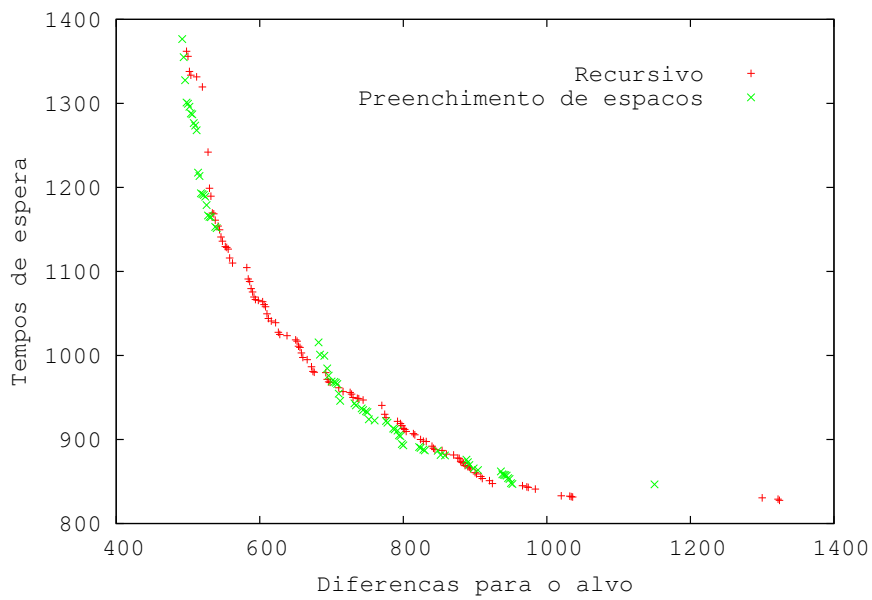


Figura 5.7: Instância com 85 localidades e padrão P_0 , 32 direções de busca.

5.3.2

Comparação das heurísticas de três fases

O próximo experimento foi realizado para comparar as estratégias de três fases utilizadas pela abordagem de resolução do bi-PAA: **3fases-biObj** é a variante que utiliza a busca local por vizinhanças após cada perturbação; **3fases-biObj + MIP** substitui a busca local pela resolução exata em todas as iterações; e **3fases-biObj + MIP_II** é a variante que utiliza um esquema híbrido combinando busca local e resolução exata com uma frequência menor que a da heurística anterior.

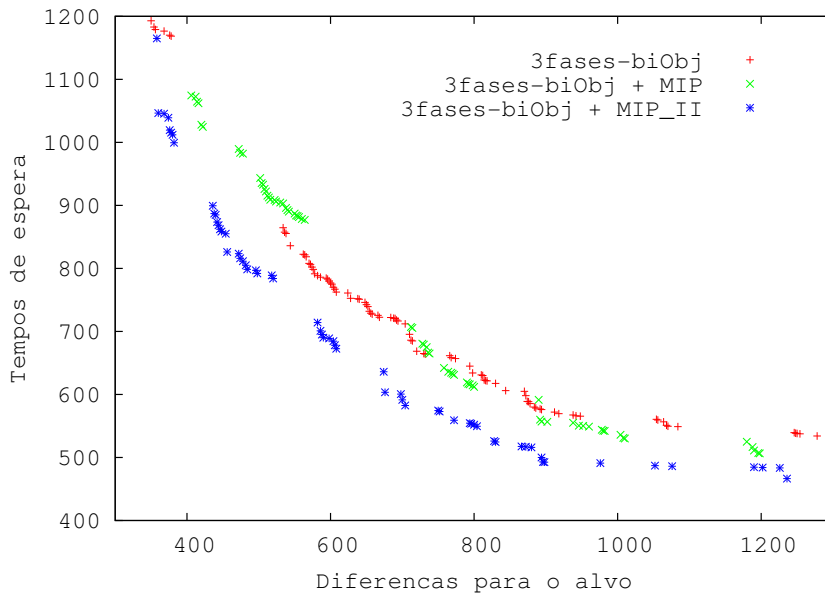


Figura 5.8: 65 localidades e padrão P_0 , 32 direções de busca.

Novamente as três instâncias teste utilizadas foram aquelas com 500 partidas e 750 árbitros e as Figuras 5.8 a 5.10 exibem as fronteiras das soluções potencialmente eficientes resultantes da aplicação de cada uma das estratégias de três fases. Neste experimento, cada figura exhibe três curvas, uma para cada estratégia de três fases. Foram sempre utilizadas 32 direções de busca.

Em todas as figuras, observa-se que a variante **3fases-biObj + MIP_II** foi a que apresentou melhores resultados, uma vez que as soluções das fronteiras associadas a esta variante dominam as soluções encontradas pelas demais variantes. Já a variante **3fases-biObj + MIP** obteve melhores resultados que a variante **3fases-biObj** em um dos casos (Figura 5.10) e resultados semelhantes para as outras duas instâncias. Assim como no caso do PAA-minTE, estes resultados mostram que a resolução exata utilizando-se o Modelo 3 realmente não é suficientemente eficiente para ser utilizada em todas iterações, como feito no caso do PAA básico com o Modelo 1.

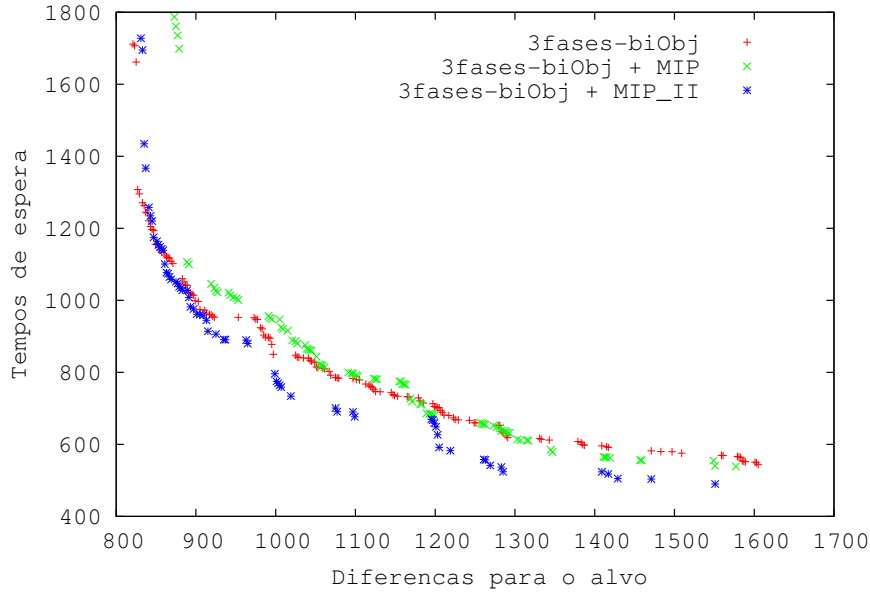


Figura 5.9: 65 localidades e padrão P_1 , 32 direções de busca.

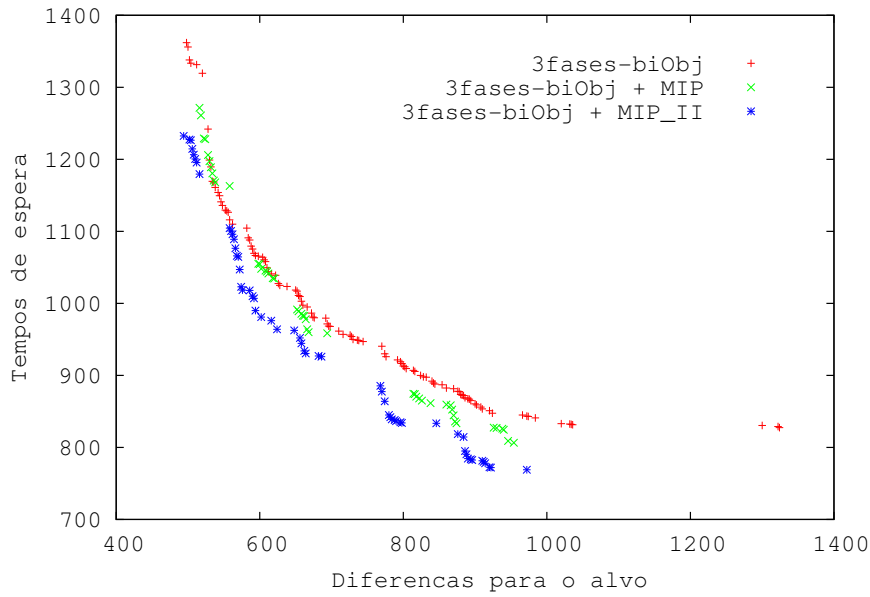


Figura 5.10: 85 localidades e padrão P_0 , 32 direções de busca.

Com o objetivo de verificar se os resultados apresentados nas Figuras 5.8 a 5.10 são afetados pelo número de direções consideradas, as Figuras 5.11 e 5.12 apresentam comparações entre as variantes **3fases-biObj** e **3fases-biObj + MIP_II** considerando-se 64 direções de busca. A variante **3fases-biObj + MIP** foi excluída deste experimento, uma vez que não obteve bons resultados para 32 direções e seu ponto fraco (a ineficiência do modelo de programação inteira que é executado em todas as iterações) não tem relação com o número de direções de busca utilizadas.

Novamente, a variante que obteve as melhores fronteiras foi **3fases-**

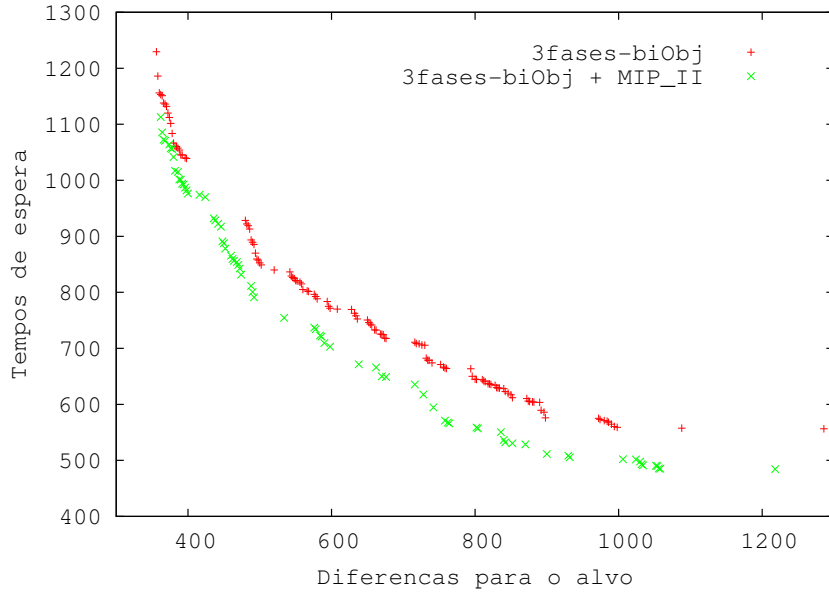


Figura 5.11: 65 localidades e padrão P_0 , 64 direções de busca.

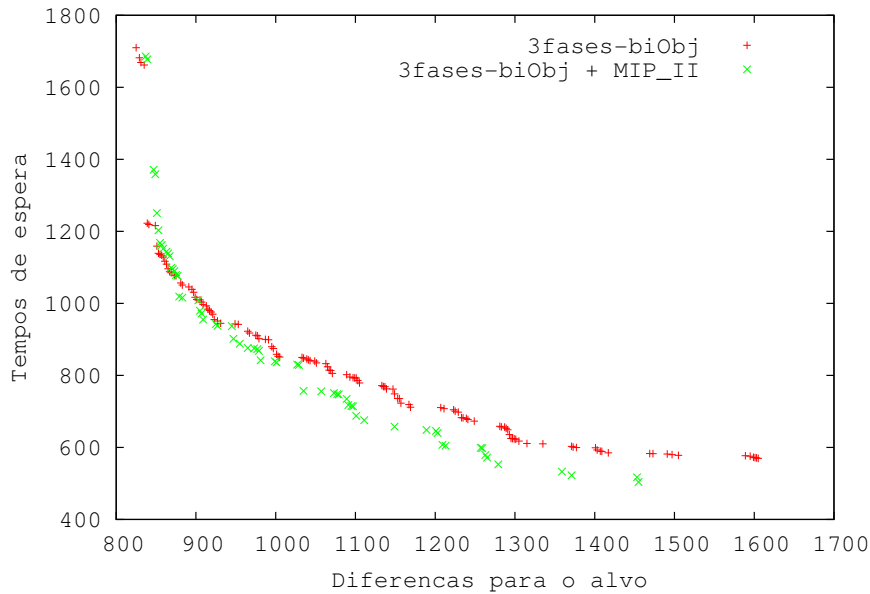


Figura 5.12: 65 localidades e padrão P_1 , 64 direções de busca.

biObj + MIP_II, sendo portanto selecionada para os demais experimentos. Com relação ao número de direções de busca utilizadas, a Figura 5.13 compara as fronteiras resultantes das aplicações da heurística 3fases-biObj + MIP_II para 32 e 64 direções de busca. Observa-se que a fronteira obtida com 64 direções apresenta-se mais homogênea, entretanto a maior parte destas soluções são dominadas por alguma solução obtida com 32 direções de busca. A conclusão que se pode tirar é que o tempo limite dado para a heurística no experimento com 64 direções não foi suficiente para que ela convergisse.

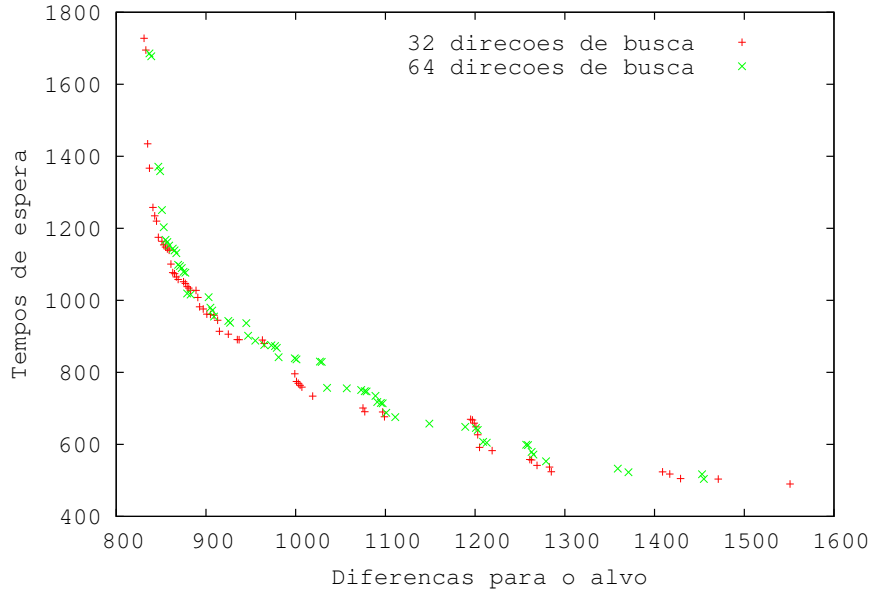


Figura 5.13: 65 localidades e padrão P_1 , heurística 3fases-biObj + MIP_II.

5.3.3

Reconexão por caminhos

Este experimento compara os resultados obtidos pelo algoritmo *Heuristica_biPAA* utilizando a estratégia de três fases com melhores resultados (3fases-biObj + MIP_II) e os resultados obtidos por uma técnica de pós-otimização através do procedimento de reconexão por caminhos proposto na Seção 5.2.3. O procedimento de pós-otimização utilizado consiste na execução iterativa do algoritmo proposto na Seção 5.2.3 para pares de soluções selecionados aleatoriamente da fronteira de soluções, até que um limite de tempo seja atingido. Para cada par selecionado de soluções, o procedimento *PathRelinking* é executado duas vezes, cada uma partindo de uma das soluções e tendo a outra como guia. O limite de tempo dado foi de 20% do tempo total gasto pelo algoritmo *Heuristica_biPAA* contando todas as execuções para todas as direções.

As Figuras 5.14 a 5.18 exibem as fronteiras obtidas pela variantes consideradas no experimento. Foram utilizadas 32 direções para obter os resultados exibidos nas Figuras 5.14 a 5.16 e 64 direções para os resultados exibidos nas Figuras 5.17 e 5.18.

Observa-se que o procedimento de reconexão por caminhos conseguiu melhorar os resultados obtidos pelo algoritmo *Heuristica_biPAA* em todos os casos, mostrando que pode ser útil na resolução do bi-PAA. A Figura 5.19 compara as fronteiras obtidas pela variante 3fases-biObj + MIP_II + PR com 32 e 64 direções de busca. Assim como na análise anterior, nota-se que a fronteira obtida com 64 direções é mais homogênea, mas o limite de tempo não

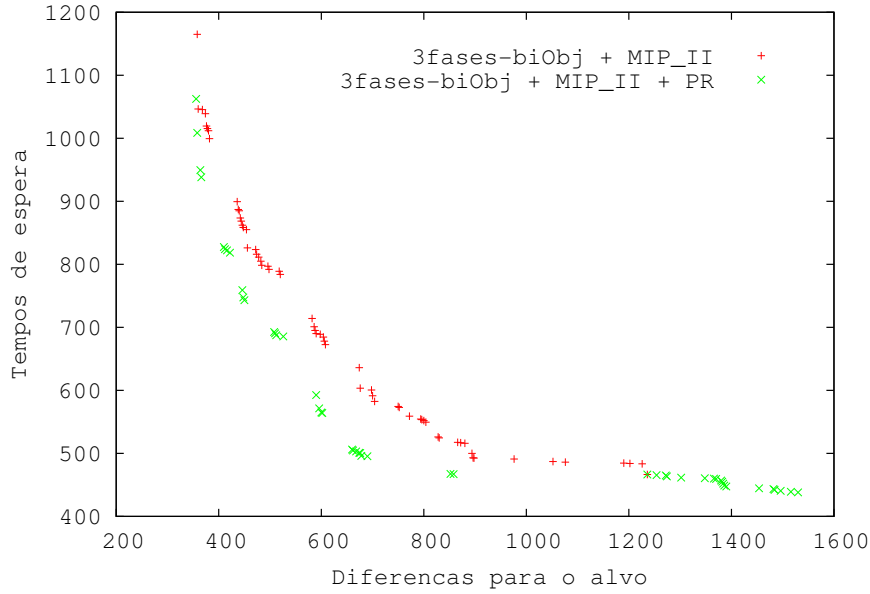


Figura 5.14: 65 localidades e padrão P_0 , 32 direções de busca, com e sem reconexão por caminhos.

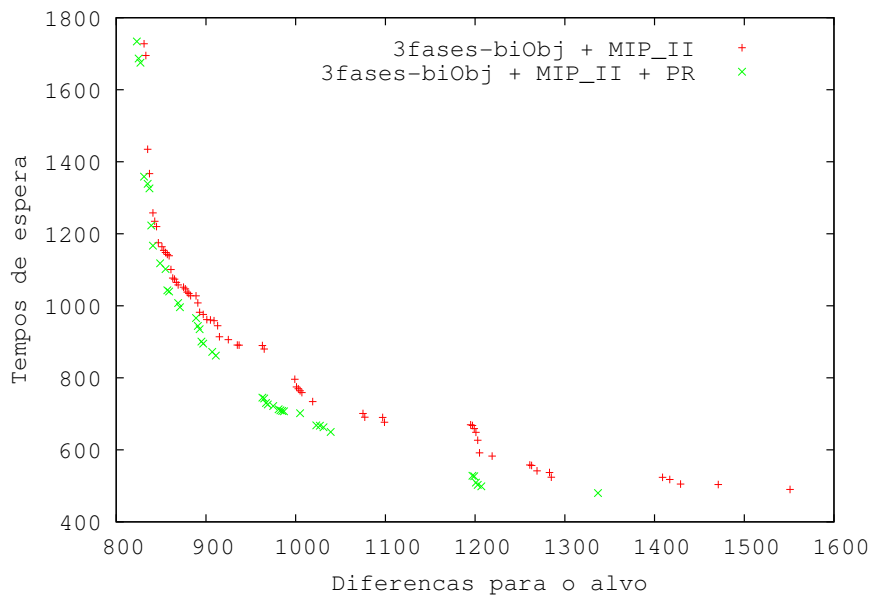


Figura 5.15: 65 localidades e padrão P_1 , 32 direções de busca, com e sem reconexão por caminhos.

foi suficiente para sua convergência, de forma que a variante que considerou 32 direções (tendo por consequência um limite maior de tempo para cada direção) obteve soluções que dominam aquelas obtidas com as 64 direções.

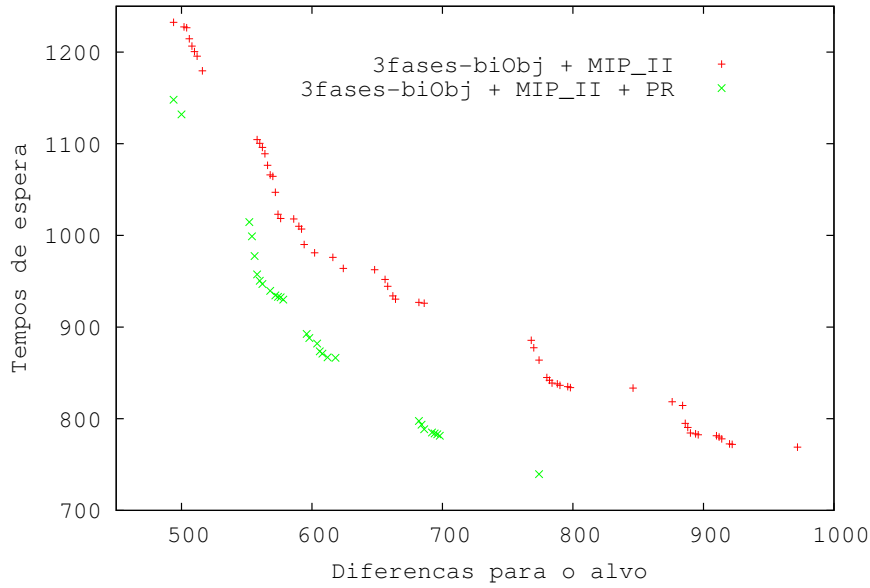


Figura 5.16: 85 localidades e padrão P_0 , 32 direções de busca, com e sem reconexão por caminhos.

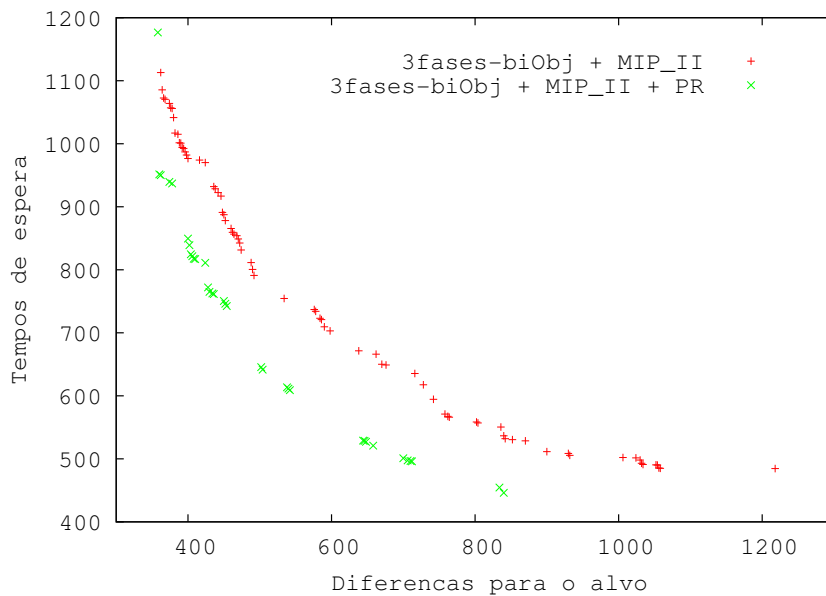


Figura 5.17: 65 localidades e padrão P_0 , 64 direções de busca, com e sem reconexão por caminhos.

5.3.4

Abordagem exata vs. aproximada

O último experimento apresentado neste capítulo mostra uma comparação entre os resultados obtidos pela resolução exata através do método dicotômico e pelas variantes heurísticas com melhor desempenho nos demais experimentos (Heurística_biPAA utilizando 3fases-biObj + MIP_II, com e sem o procedimento de reconexão por caminhos).

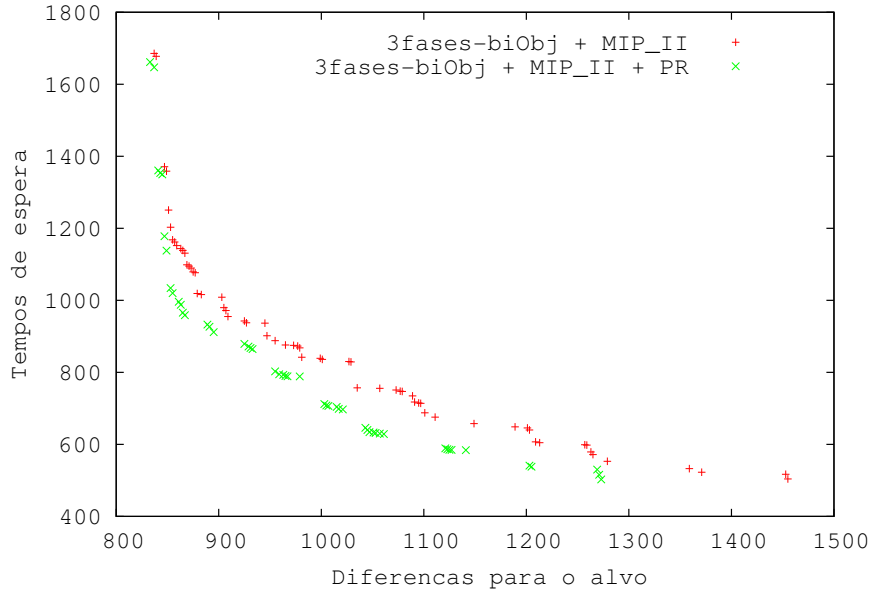


Figura 5.18: 65 localidades e padrão P_1 , 64 direções de busca, com e sem reconexão por caminhos.

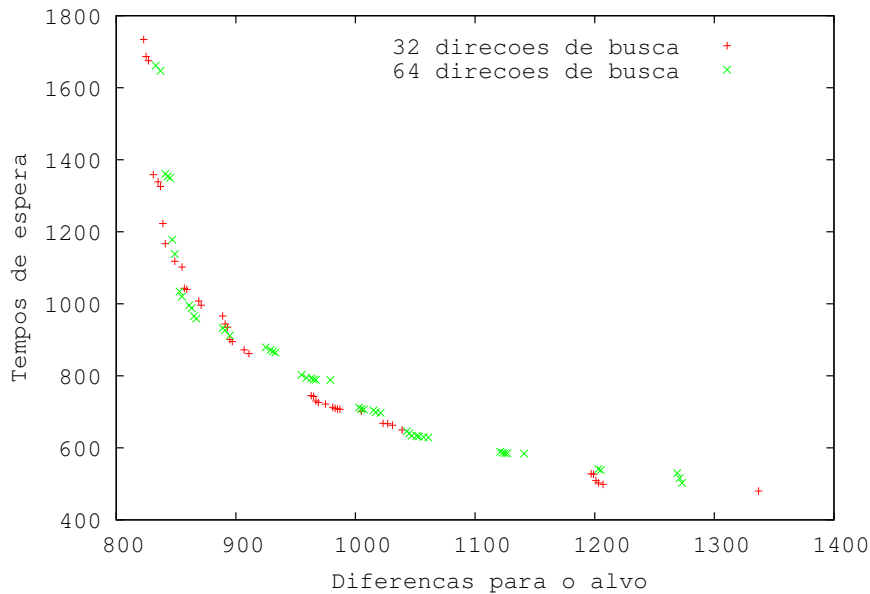


Figura 5.19: 65 localidades e padrão P_1 , 3fases-biObj + MIP_II + PR.

São apresentados os resultados para as instâncias com 50 e 60 partidas. Foram utilizados também 32 e 64 direções de busca. As Figuras 5.20 a 5.24 exibem, além das fronteiras para as duas variantes heurísticas (3fases-biObj + MIP_II, 3fases-biObj + MIP_II + PR), a fronteira de Pareto obtida de maneira exata pelo método dicotômico (*Exato*) e também as aproximações da fronteira após a fase de construção e após a primeira fase de busca local. Ressalta-se que no caso das estratégias de três fases utilizadas neste experimento (3fases-biObj + MIP_II e 3fases-biObj + MIP_II + PR), a

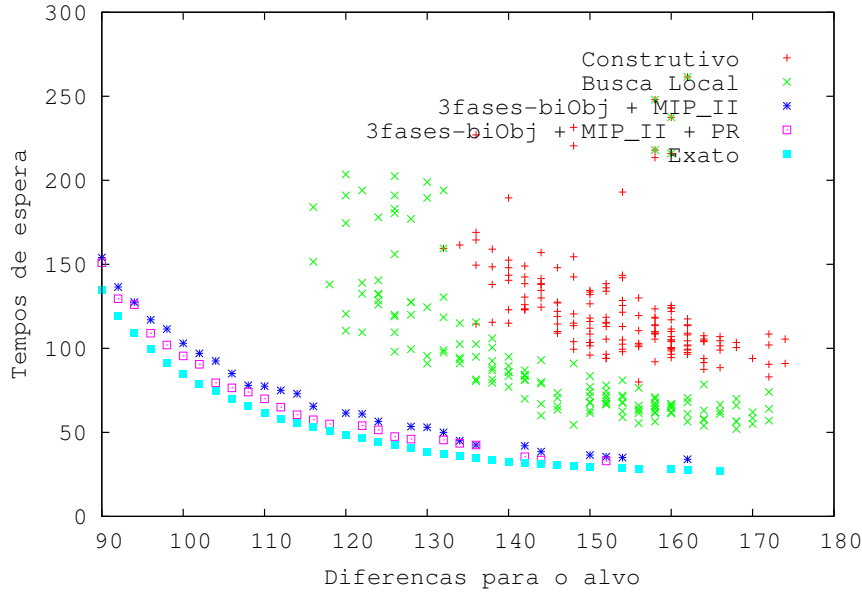


Figura 5.20: Instância com 50 partidas, 100 árbitros, 6 localidades e padrão P_0 , 32 direções de busca.

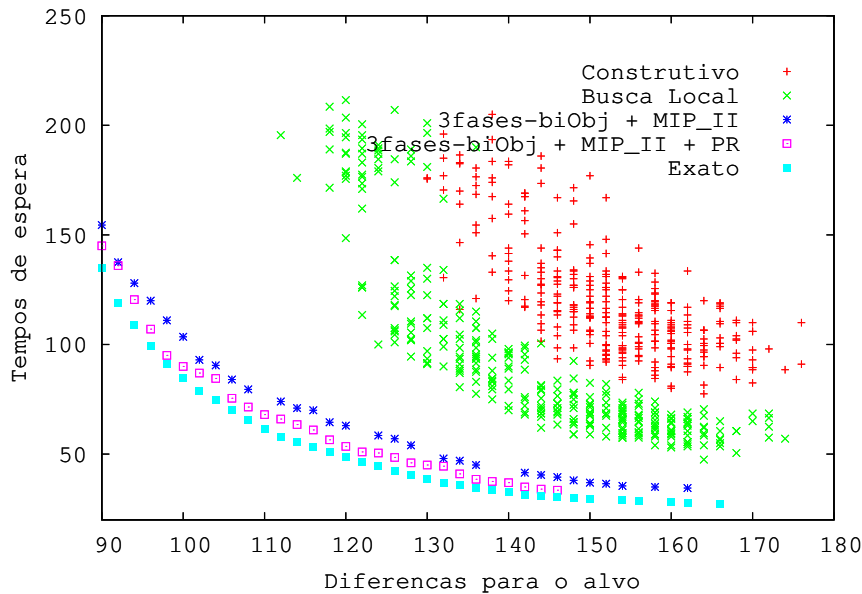


Figura 5.21: Instância com 50 partidas, 100 árbitros, 6 localidades e padrão P_0 , 64 direções de busca.

fase de busca local foi substituída pela resolução exata de sub-problemas de programação inteira. As Figuras 5.25 e 5.26 comparam as fronteiras da variante com reconexão por caminhos utilizando 32 e 64 direções com a fronteira de Pareto computada de maneira exata.

Estes resultados mostram, não somente a evolução das soluções encontradas pela heurística em cada uma de suas fases, como também a proximidade da fronteira resultante em relação à fronteira de Pareto computada de maneira

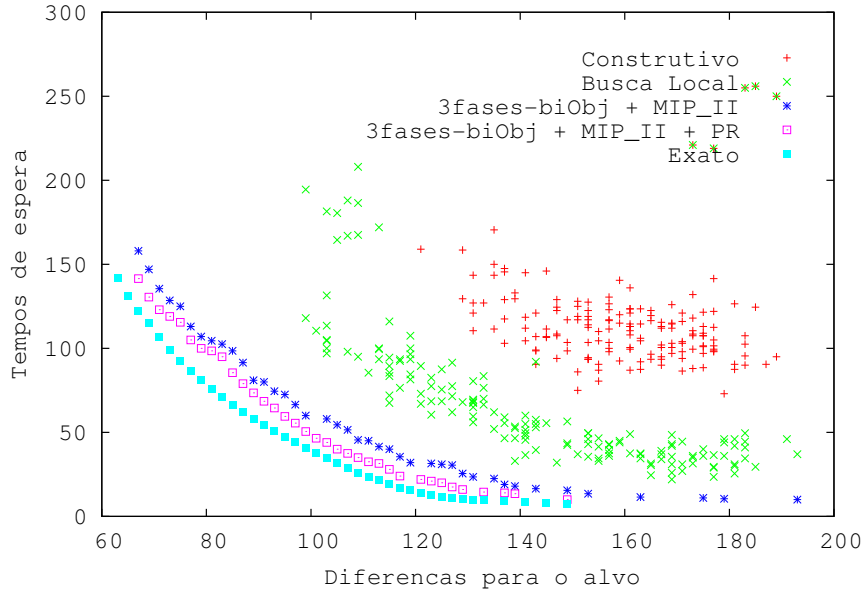


Figura 5.22: Instância com 60 partidas, 120 árbitros, 6 localidades e padrão P_0 , 32 direções de busca.

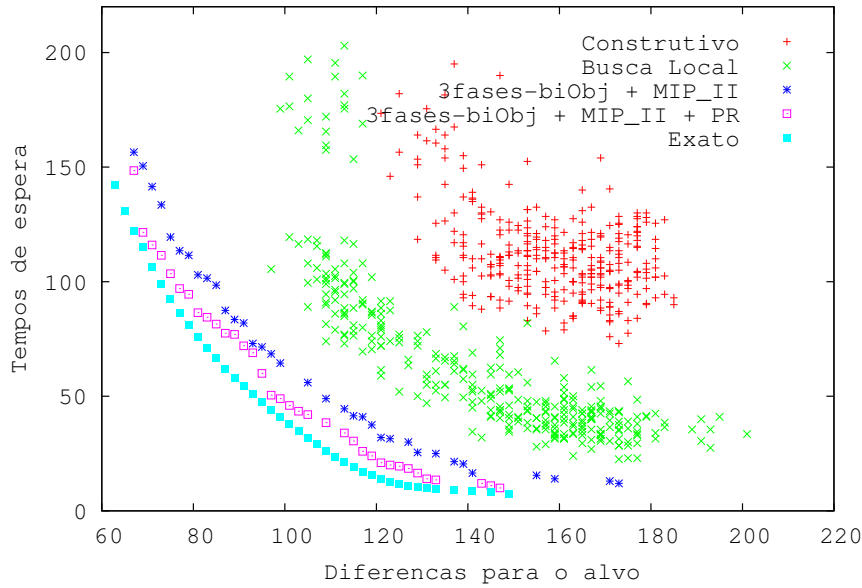


Figura 5.23: Instância com 60 partidas, 120 árbitros, 6 localidades e padrão P_0 , 64 direções de busca.

exata. Observa-se novamente a melhora proporcionada pelo procedimento de pós-otimização através da reconexão por caminhos. Na comparação entre as fronteiras obtidas com diferentes números de direções, novamente não se pode chegar à conclusão de que uma escolha é melhor que a outra. Neste caso, entretanto, a fronteira obtida com 32 direções se apresentou tão homogênea quanto aquela obtida com 64 direções. Este resultado indica que possivelmente o número ideal de direções utilizadas varia em função do tamanho das instâncias.

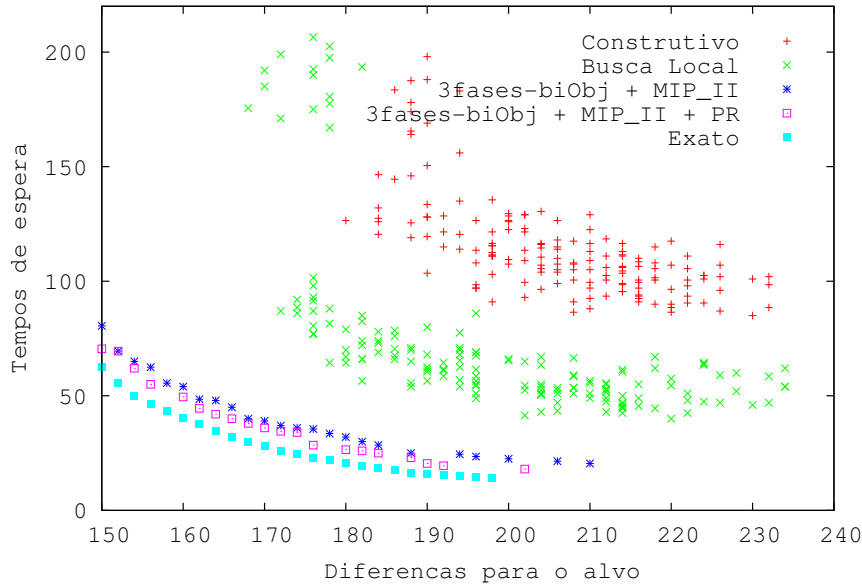


Figura 5.24: Instância com 60 partidas, 120 árbitros, 6 localidades e padrão P_1 , 32 direções de busca.

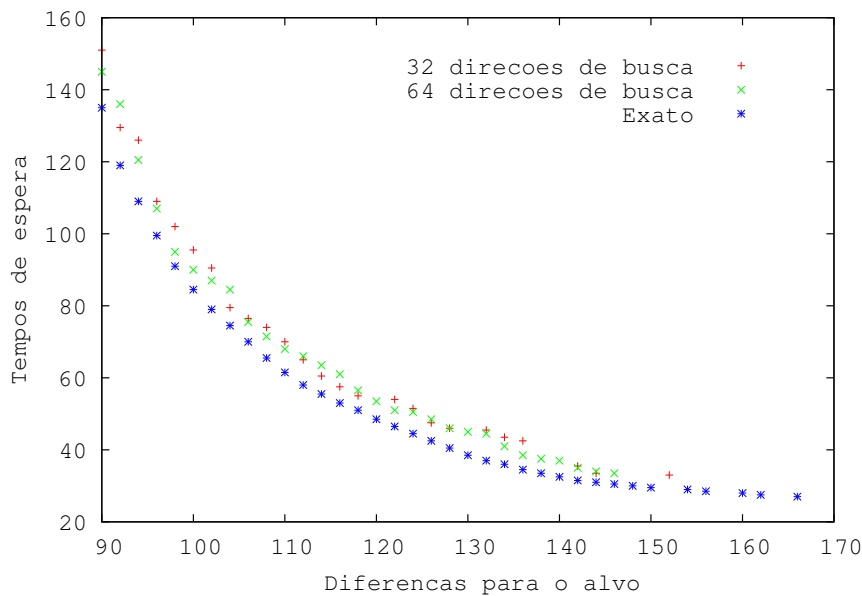


Figura 5.25: Instância com 50 partidas, 100 árbitros, 6 localidades e padrão P_0 , heurística 3fases-biObj + MIP_II + PR.

5.4 Conclusões

Este capítulo propôs abordagens de resolução exata e aproximada para uma variante bi-objetivo do PAA que considera simultaneamente os dois objetivos tratados nos capítulos anteriores.

O conjunto mínimo completo de soluções eficientes foi computado para instâncias de médio porte através do método dicotômico. Foi apresentada uma

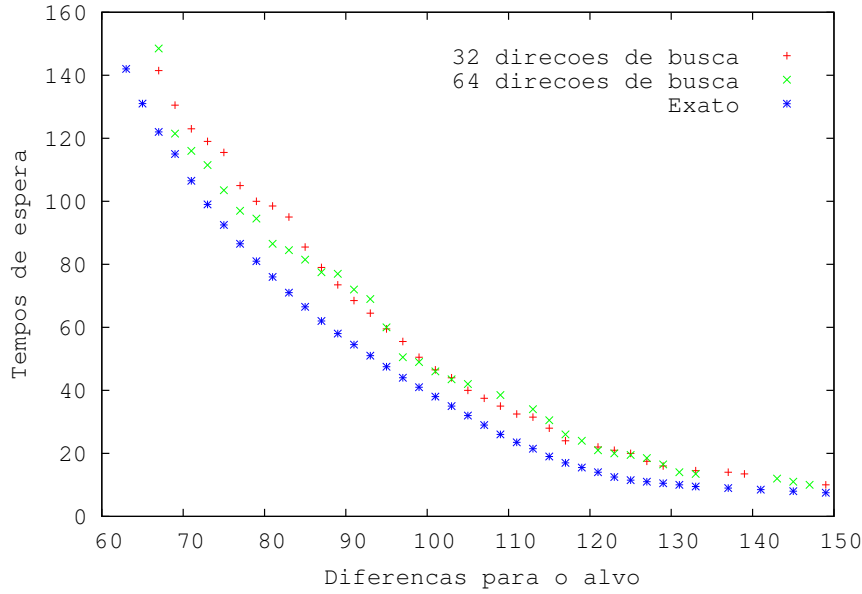


Figura 5.26: Instância com 60 partidas, 120 árbitros, 6 localidades e padrão P_0 , heurística *3fases-biObj* + *MIP_II* + *PR*.

abordagem aproximada promissora, baseada nas heurísticas de três fases propostas para as variantes mono-objetivo e utilizando estratégias híbridas de busca local com vizinhanças e resolução exata por programação inteira. Diversos experimentos computacionais foram utilizados na comparação dos métodos propostos e os resultados mostraram a eficácia da abordagem proposta na resolução de instâncias de médio e grande portes. Comparações entre as fronteiras de Pareto computadas de maneira exata pelo método dicotômico para instâncias de médio porte e as fronteiras aproximadas obtidas pela abordagem heurística mostraram a qualidade das soluções obtidas pela heurística. Observa-se também pelos resultados a melhoria progressiva das soluções após a atuação de cada componente da heurística.

Foi proposto também um procedimento de reconexão por caminhos que foi utilizado como uma pós-otimização e melhorou a qualidade da fronteira encontrada com as demais abordagens. Heurísticas de reconexão por caminhos têm se mostrado importantes ferramentas na resoluções de problemas multi-objetivo, de forma que a investigação de outras formas de utilização do procedimento proposto para o bi-PAA constitui uma interessante linha para pesquisas futuras.