

4 Resultados da IPC-4

Este capítulo apresenta o benchmark criado sobre instâncias do Pipesworld 2 e comenta os resultados obtidos pelos planejadores de propósito geral sobre este benchmark durante a *International Planning Competition 2004* - IPC-4. A Seção 4.1 descreve os eventos IPC, a seção 4.2 apresenta as características das instâncias utilizadas no benchmark da competição, e como estas foram geradas. A Seção 4.3 apresenta e analisa os resultados obtidos, enquanto a Seção 4.4 apresenta as conclusões e motivações para os capítulos seguintes.

4.1 A Competição Internacional de Planejamento

Atualmente, a construção de mecanismos eficientes de resolução de problemas gerais é uma área importante na pesquisa de planejamento em IA. O desempenho destes mecanismos, implementados por planejadores de propósito geral, é medido através de testes destes sistemas em exemplos de instâncias dos problemas a serem resolvidos. Tipicamente, é considerado o “melhor” algoritmo em uma determinada época aquele que consegue resolver estas instâncias de maneira mais eficiente. Como nenhum mecanismo irá resolver todas as instâncias de um problema difícil, um dos aspectos cruciais no desenvolvimento deste campo de pesquisa está na definição de quais tipos de instâncias são utilizadas para esta avaliação.

Cada vez mais, as instâncias da Competição Internacional de Planejamento ou IPC têm sido utilizadas para este fim. Desta forma, os benchmarks da IPC são atualmente um dos instrumentos mais importantes nesta área de pesquisa.

Durante a quarta edição da IPC, o comitê organizador investiu um esforço considerável na criação de um benchmark que considerasse os seguintes aspectos: orientados a aplicações, com estrutura diversa e apropriado para pesquisa básica. Estes aspectos são detalhados a seguir.

O campo de planejamento em IA tem sido freqüentemente criticado por uma certa obsessão com problemas de brinquedo, ou *toy problems*. Recentemente, com a melhoria de desempenho dos planejadores, foi possível trabalhar com exemplos mais realistas. A IPC-4 procurou dar um passo adiante neste aspecto, selecionando aplicações reais para inspirar os problemas de planejamento do benchmark, e adicionando complexidade aos problemas clássicos utilizados nas versões anteriores da competição.

Em relação à estrutura diversa, é conhecido que o desempenho de planejadores de busca heurística depende fortemente da qualidade desta heurística. Resultados obtidos em [19] sugerem que o bom desempenho de alguns destes planejadores nas edições anteriores da IPC foi devido à similaridades na estrutura dos problemas mais tradicionais que formaram o benchmark utilizado nestas edições. Isto não quer dizer que os planejadores que utilizam técnicas de busca heurística não sejam úteis, mas alerta para o risco de que o benchmark possa favorecer uma determinada forma de solução. Na montagem do benchmark da IPC-4, tomou-se o cuidado de selecionar problemas que intuitivamente possuem estruturas distintas.

Em relação ao uso em pesquisa básica, o IPC-4 procurou gerar codificações em PDDL reduzíveis às construções da linguagem STRIPS [7]. Mas, ao invés de simplesmente ignorar algumas características do problema, estas foram “compiladas” para se restringir ao STRIPS. Como exemplo de compilação para STRIPS podemos citar o processo de modelagem da tancagem como *slots*, descrito na Seção 3.3.1.

Pelas suas características, o domínio Pipesworld 2 apresentado na Seção 3.3 foi um dos domínios selecionados para o IPC-4, juntamente com outros que tratavam de aplicações como controle de tráfego aéreo e restauração de energia em redes elétricas.

4.2

O benchmark Pipesworld utilizado no IPC-4

O benchmark baseado no Pipesworld 2 utilizado na IPC-4 define várias instâncias que possuem um grau incremental de dificuldade. As instâncias mais fáceis tem menor tamanho, em termos de topologia da rede e número de bateladas, e podem ignorar algumas restrições. As instâncias mais difíceis são maiores, e levam em consideração todas as restrições apresentadas na Seção 3.3.

O benchmark é composto por cinco conjuntos diferentes, detalhados nas seções seguintes. A motivação para a criação destes conjuntos é que nem

todos os planejadores de propósito geral podem lidar com efeitos temporais, e construindo conjuntos com e sem restrição de tancagem é possível avaliar o quanto esta restrição em particular torna o problema mais difícil.

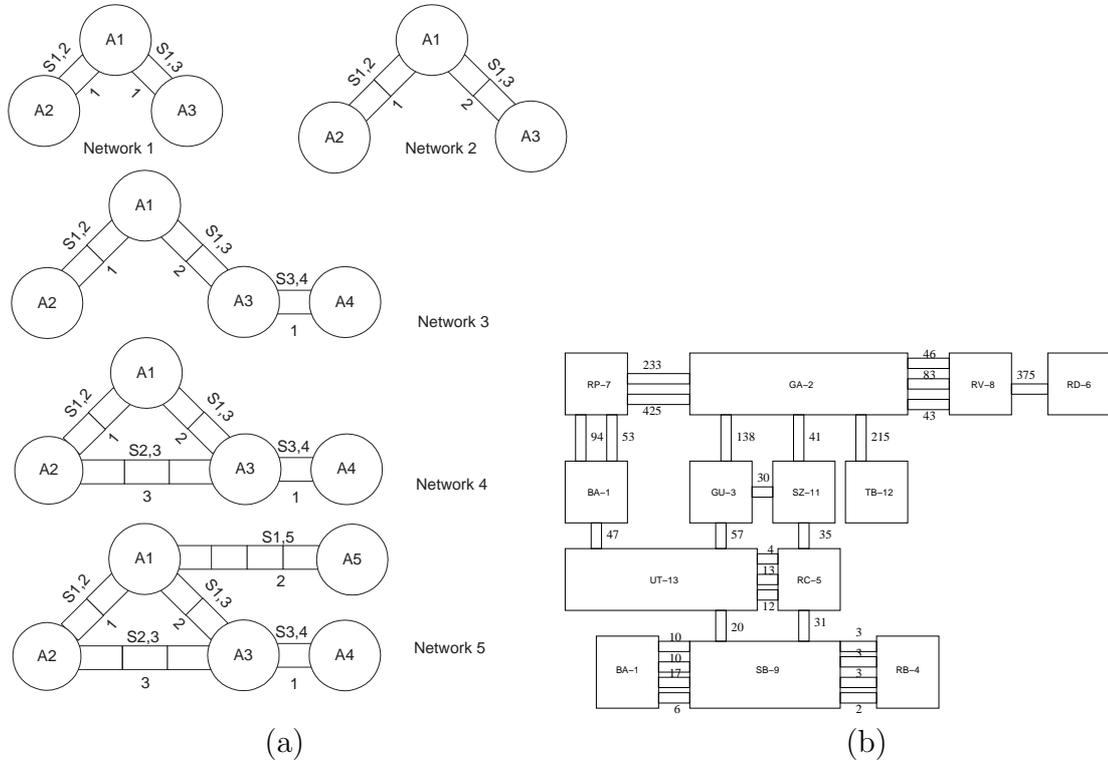


Figura 4.1: Topologias de rede utilizadas no IPC-4 (a) e uma rede real (b)

Cada conjunto, com exceção do conjunto que considera prazos, é composto por 50 instâncias distintas. As instâncias são geradas aleatoriamente, escalando o número total de bateladas e o número de bateladas não proteláveis. Foram geradas 10 instâncias para cada topologia de rede mostrada na Figura 4.1 - (a). A mesma figura mostra uma rede real para efeitos de comparação. Podemos observar que as redes utilizadas no benchmark são mais simples do que o exemplo real, mas não a ponto de serem triviais. Os volumes dos dutos nesta figura são dados em unidades de $100m^3$.

Todas as instâncias trabalham com 5 produtos e consideram a restrição de interface (Seção 3.2.1). Estes produtos e suas respectivas interfaces, mostrados na Figura 4.2, representam um caso real¹.

¹O conjunto completo das instâncias pode ser obtido no endereço <http://ls5-www.cs.uni-dortmund.de/edelkamp/ipc-4>

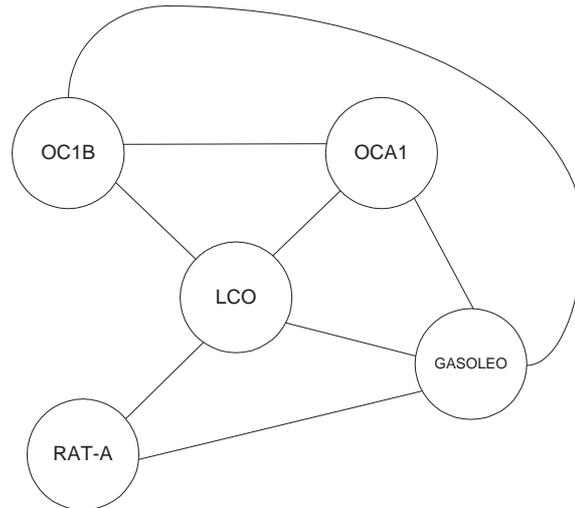


Figura 4.2: Grafo de interface de produtos do benchmark do IPC-4

4.2.1

Sem tancagem, sem tempo

O conjunto *no-tankage* é o mais simples do benchmark. Ele não leva em consideração nenhum efeito temporal, ou seja, a vazão de todos os dutos é a mesma. Além disso, as bateladas não possuem prazo de entrega, e a capacidade de armazenamento é ilimitada para todos os produtos em todas as áreas.

4.2.2

Com tancagem, sem tempo

O conjunto *tankage* estende o conjunto *no-tankage* para considerar a restrição de tancagem nas áreas. Para garantir que todas as instâncias fossem viáveis e ao mesmo tempo desafiadoras, foram geradas diversas instâncias com o valor das tancagens para cada produto e em cada área sendo definido de forma aleatória dentro do intervalo de 0 até um valor máximo suficiente para armazenar todas as bateladas do produto existentes na rede. O planejador de propósito geral MIPS [25] foi utilizado nestas instâncias, e foram selecionadas para o benchmark apenas instâncias resolvidas, garantindo desta forma a viabilidade. Algumas instâncias não foram solucionadas pelo MIPS mesmo com um tempo de execução de vários dias, para estas foi utilizado o valor máximo de tancagem, o que também garante a viabilidade.

4.2.3

Sem tancagem, com tempo

As instâncias do conjunto *temporal-no-tankage* são idênticas às do conjunto *no-tankage* (Seção 4.2.1) em termos de topologia da rede e características das bateladas. Elas no entanto são estendidas para considerar ações que possuem duração distintas, em função de vazões diferentes para os dutos.

A capacidade de armazenamento nas áreas é ilimitada, e as restrições de prazo não são consideradas. Ou seja, qualquer solução válida para uma instância do conjunto *tankage* é também válida para este conjunto, já que as ações com duração neste caso modificam apenas o tempo de chegada das bateladas ao seu destino.

4.2.4

Com tancagem, com tempo

As instâncias do conjunto *temporal-tankage* são idênticas às do conjunto *tankage* (Seção 4.2.2) em termos de topologia de rede, características das bateladas e capacidade máxima de armazenamento nas áreas. Da mesma forma descrita na Seção anterior, elas são estendidas para considerar ações temporais mas não incluem as restrições de prazo.

4.2.5

Sem tancagem, com tempo e prazos

As instâncias do conjunto *temporal-no-tankage-deadlines* adicionam a restrição de prazo ao conjunto *temporal-no-tankage* (Seção 4.2.3). A adição dos prazos foi um desafio na geração do benchmark, já que gerar instâncias desafiadoras mas comprovadamente viáveis não é trivial.

O processo de geração das instâncias neste conjunto foi o seguinte. Inicialmente executamos o MIPS nas instâncias temporais e sem tancagem. Nas instâncias que foram resolvidas, foram adicionados prazos para cada batelada não protelável. Os prazos foram escolhidos aleatoriamente dentro do intervalo entre o tempo de chegada da batelada no plano gerado pelo MIPS e o tempo final do mesmo plano. As instâncias não resolvidas pelo MIPS foram deixadas de fora do benchmark.

4.3

Análise dos resultados

Esta Seção analisa os resultados obtidos pelos planejadores na IPC-4 especificamente para as instâncias do Pipesworld. Os resultados são analisados separadamente para cada um dos conjuntos citados na Seção anterior.

Para cada conjunto, são apresentados gráficos que capturam o desempenho dos planejadores em termos de tempo de execução e, para os planejadores sub-ótimos, qualidade do plano obtido. O desempenho em termos do tempo de execução é dividido em dois gráficos, cada um com um grupo de planejadores, para facilitar a visualização do resultado dos conjuntos não temporais. Neste conjunto, o número de participantes foi muito grande, e a visualização em um único gráfico ficaria prejudicada. A qualidade do plano é medida ou em termos do número de ações do mesmo ou em seu makespan. Durante a competição, cada planejador pôde escolher o critério sobre o qual a qualidade do plano resultante seria avaliada.

De forma geral, o desempenho dos planejadores melhorou bastante em relação à edição anterior da competição, com os vencedores resolvendo diversas instâncias que os organizadores consideravam inviáveis na prática, tomando como referência o desempenho dos vencedores da IPC-3 para as mesmas.

Todos os resultados foram obtidos com a seguinte configuração de *hardware*: IBM PC, Pentium IV com 3 GHz e 1 GB de RAM. O tempo limite para resolução de cada uma das instâncias foi de 30 minutos.

4.3.1

Sem tancagem, sem tempo

Esta Seção analisa os resultados para o conjunto *no-tankage*, descrito na Seção 4.2.1.

As Figuras 4.3 e 4.4 mostram o tempo de execução para os 13 planejadores sub-ótimos. Descontinuidades nas curvas dos planejadores são geradas no local correspondente às instâncias não resolvidas. Apenas os planejadores SGPlan [31] e YAHSP[32] resolveram todas as instâncias deste conjunto. O planejador MacroFF [33] foi o mais rápido em muitas das instâncias, mas não resolveu muitas outras também.

A Figura 4.5 mostra o tempo de execução para os planejadores ótimos no conjunto. Podemos notar que estes planejadores ainda não escalam de forma satisfatória, resolvendo apenas as instâncias de menor dimensão. Esta

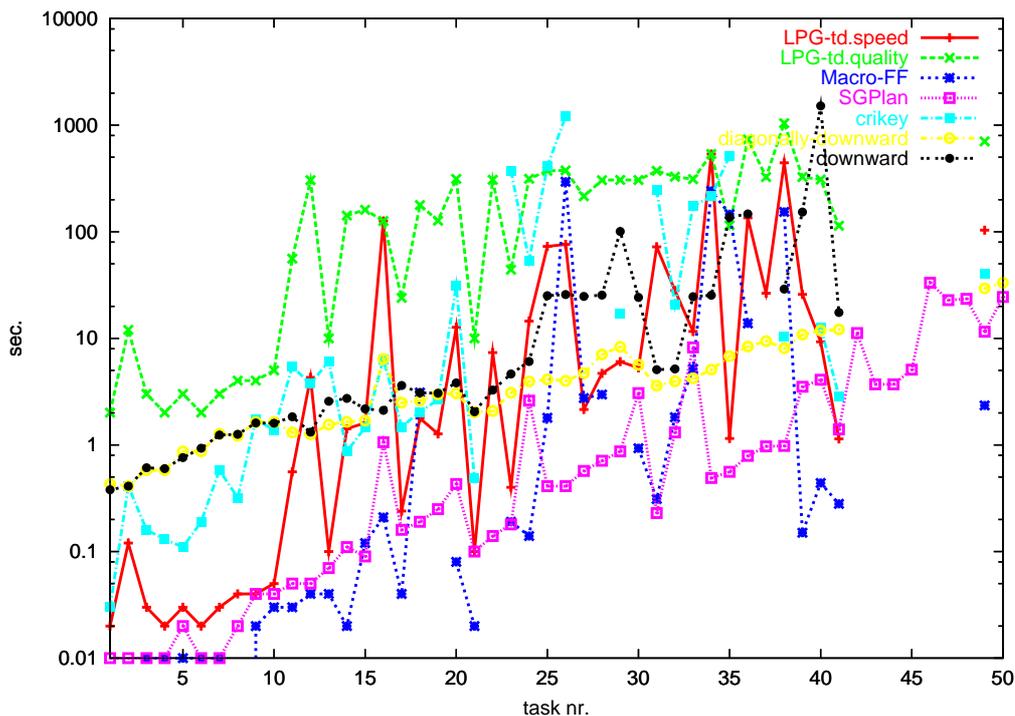


Figura 4.3: *notankage*, tempo de execução, sub-ótimos, grupo 1

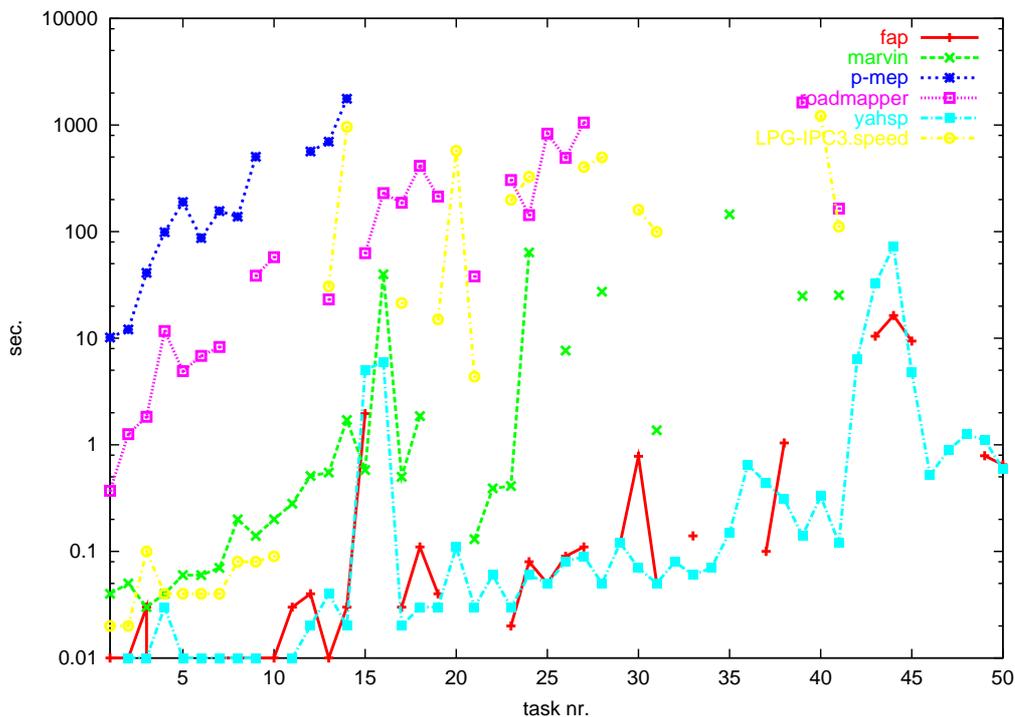


Figura 4.4: *notankage*, tempo de execução, sub-ótimos, grupo 2

dificuldade na otimização do processo reforça a decisão de definirmos o nosso problema como de planejamento, e não de escalonamento, conforme descrito

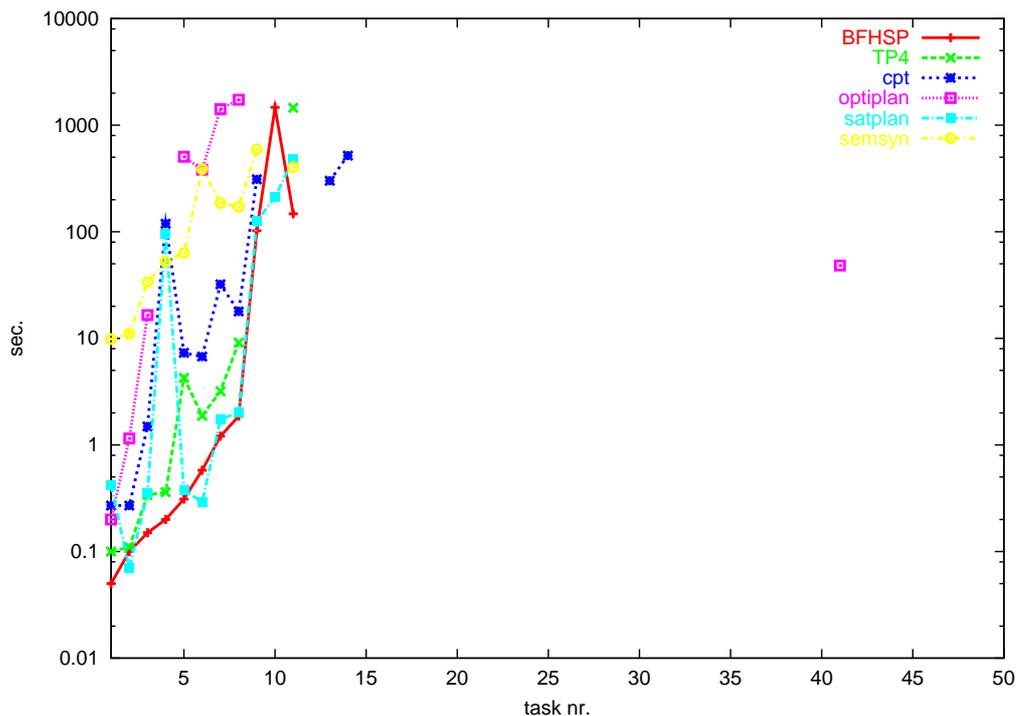


Figura 4.5: *notankage*, tempo de execução, ótimos

na Seção 2.3.

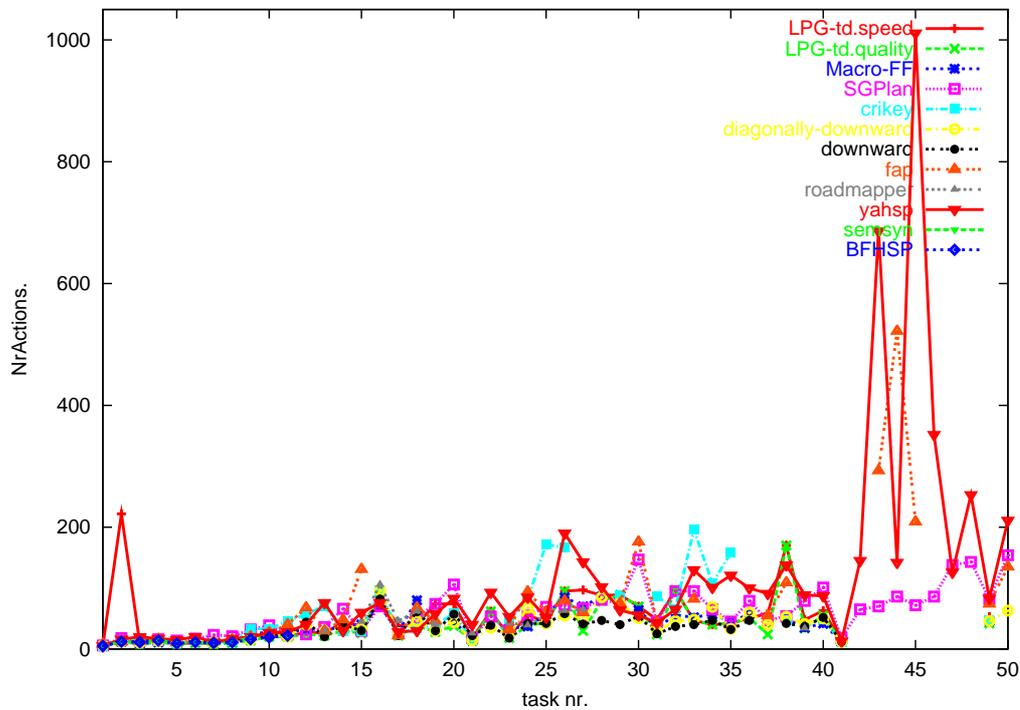


Figura 4.6: *notankage*, número de ações

A Figura 4.6 compara a qualidade dos planos obtidos, em função

do número de ações no mesmo. Podemos notar que, com exceção do YAHSP e do FAP [34], que geraram planos desnecessariamente longos para algumas instâncias, a qualidade dos planos é de certa forma equivalente, mesmo quando comparamos os planejadores sub-ótimos com os planejadores ótimos.

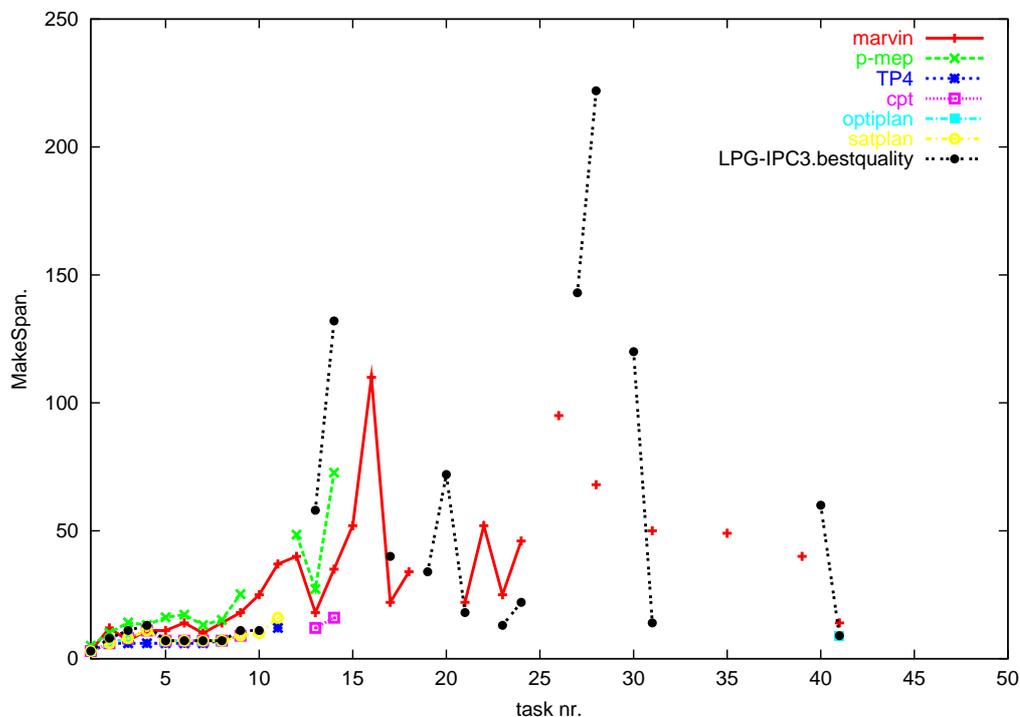


Figura 4.7: *notankage*, makespan

A Figura 4.7 compara a qualidade dos planos obtidos, em função de seu makespan. Podemos observar que os planejadores que adotaram o makespan como critério de avaliação da qualidade do plano tiveram um desempenho bem pior do que os que optaram pelo número de ações, em relação ao número de instâncias resolvidas. Isto é esperado, já que estes planejadores trabalham tentando minimizar o makespan, e esta função objetivo é bem mais complexa do que a minimização do número de ações.

4.3.2 Com tancagem, sem tempo

Esta Seção analisa os resultados para o conjunto *tankage*, descrito na Seção 4.2.2.

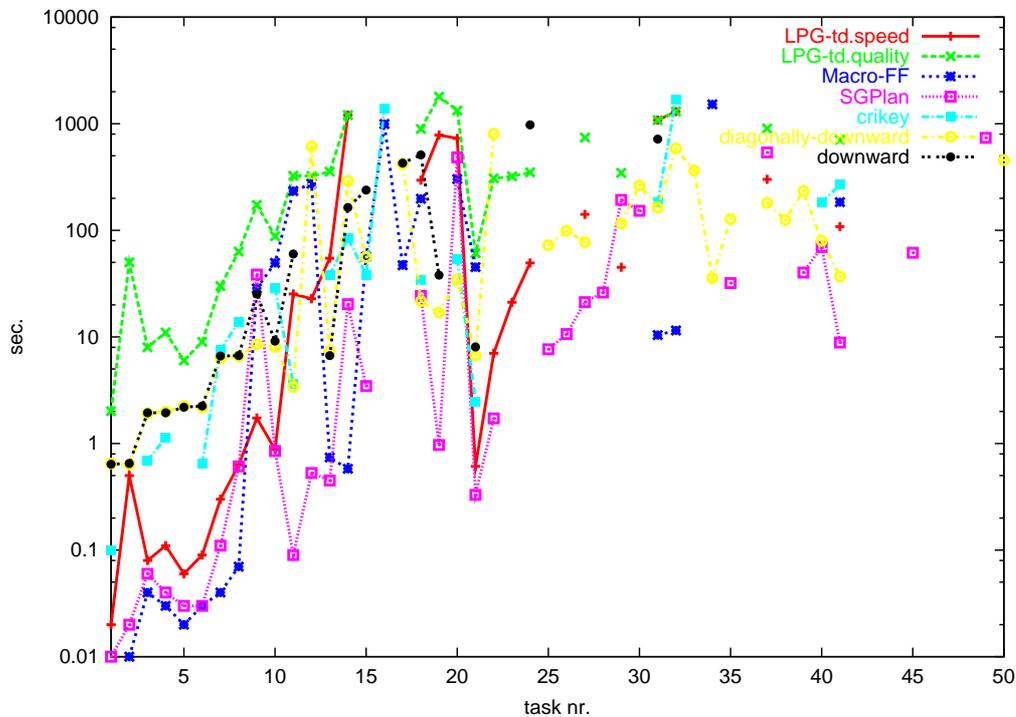


Figura 4.8: *tankage*, tempo de execução, sub-ótimos, grupo 1

As Figuras 4.8 e 4.9 mostram o tempo de execução para os 13 planejadores sub-ótimos. Inicialmente, comparando com as Figuras 4.3 e 4.4 observamos que, como esperado, a inclusão da restrição de tancagem torna o problema mais difícil para os planejadores, com uma redução significativa do número de instâncias resolvidas e do tempo necessário para a geração dos planos nestas instâncias. Em relação ao número de instâncias resolvidas, o *Fast Diagonally Downward*, FDD [35] e o YAHSP destacaram-se.

As mesmas observações feitas na Seção 4.3.1 se aplicam às Figuras 4.10, 4.11 e 4.12.

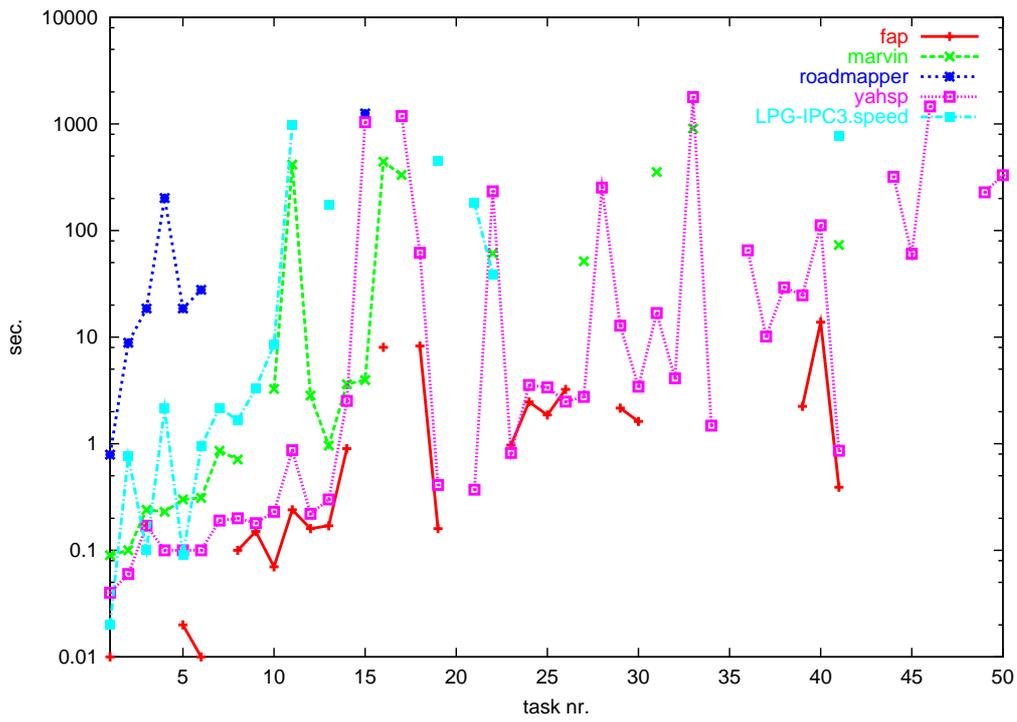


Figura 4.9: *tankage*, tempo de execução, sub-ótimos, grupo 2

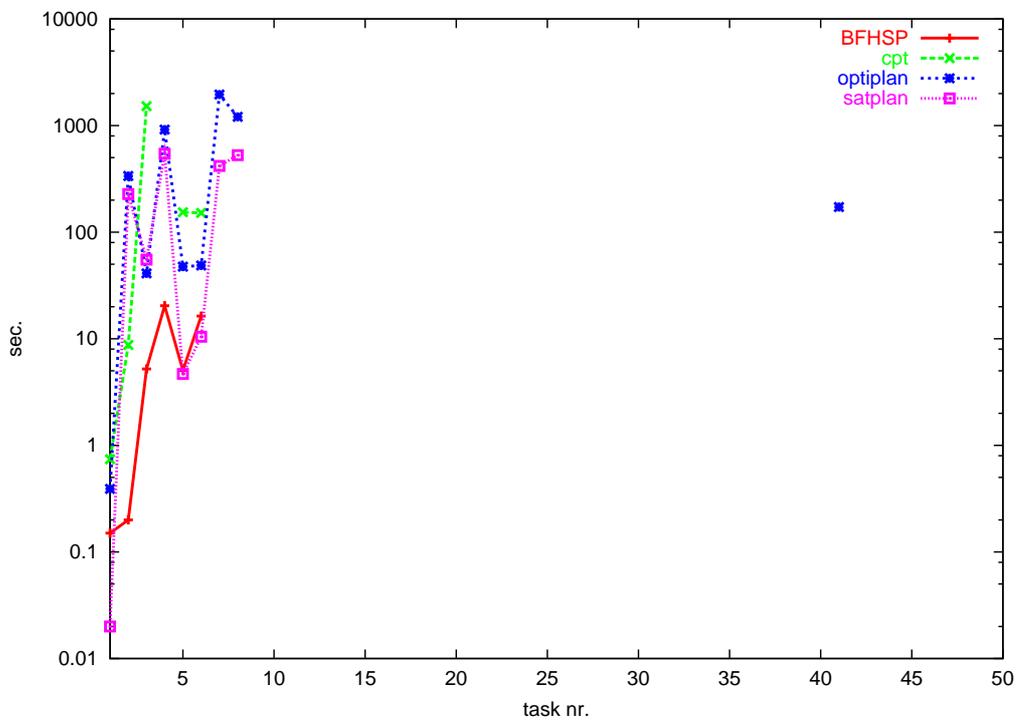


Figura 4.10: *tankage*, tempo de execução, ótimos

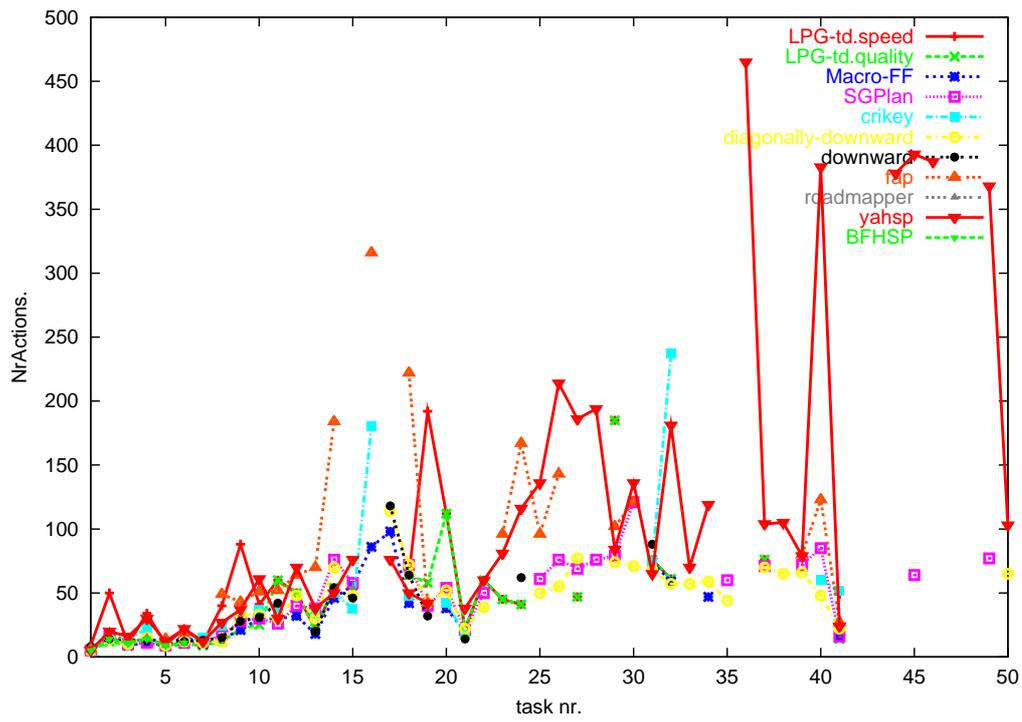


Figura 4.11: *tankage*, número de ações

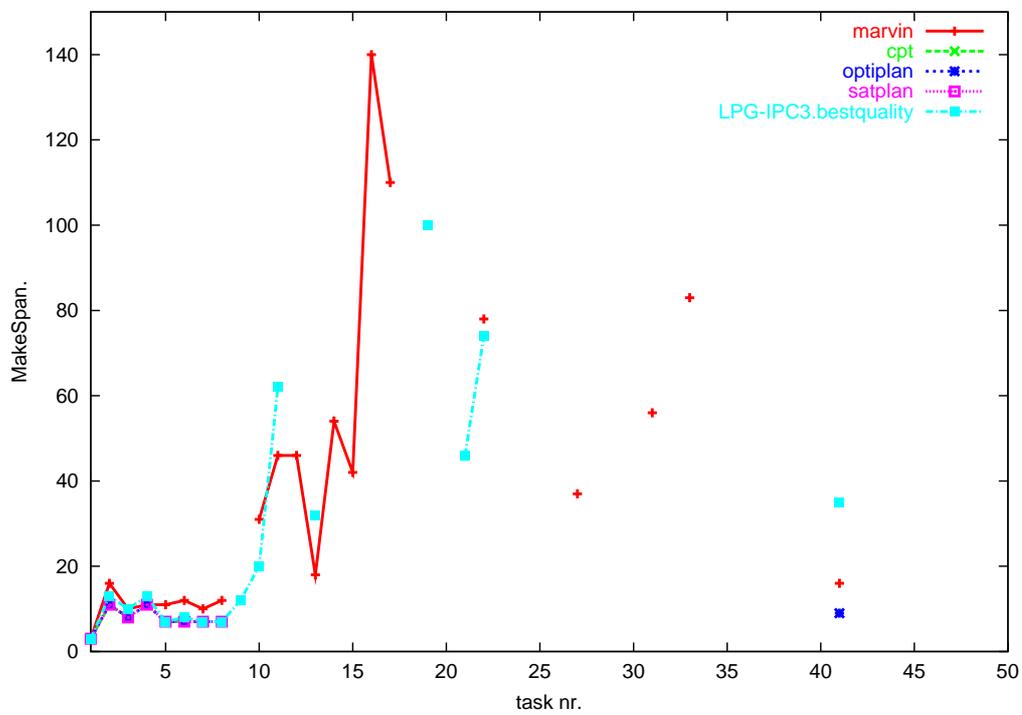


Figura 4.12: *tankage*, makespan

4.3.3 Sem tancagem, com tempo

Esta Seção analisa os resultados para o conjunto *temporal-no-tankage*, descrito na Seção 4.2.3.

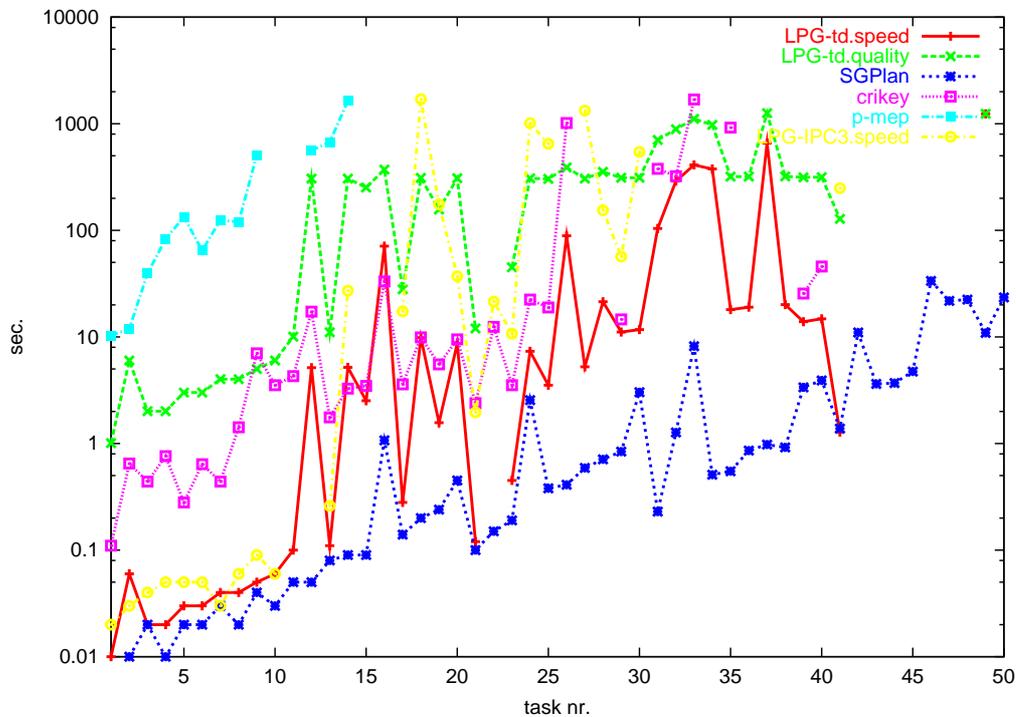


Figura 4.13: *temporal-no-tankage*, tempo de execução, sub-ótimos

As Figuras 4.13 e 4.14 mostram o tempo de execução dos planejadores sub-ótimos e ótimos. A redução no número de planejadores mostrados quando comparamos com os gráficos das duas últimas seções é função do menor número de planejadores que suportam ações com durações distintas.

Como observado na Seção 4.2.3, todo plano válido para o conjunto que não considera nem tancagem nem durações distintas para as ações é também válido para estas instâncias. Por conta disto os planejadores que utilizam como métrica a ser minimizada o número total de ações do plano resultante tiveram um desempenho idêntico aos gráficos mostrados na Seção 4.3.1, em relação ao tempo de execução e ao número de ações. Já os planejadores que minimizavam o makespan tiveram seu desempenho modificado.

Como exemplo de planejador que teve o mesmo tipo de comportamento para as instâncias temporais e não temporais podemos citar o SGPLAN, e como um que teve o seu comportamento modificado podemos citar o LPG-IPC3.

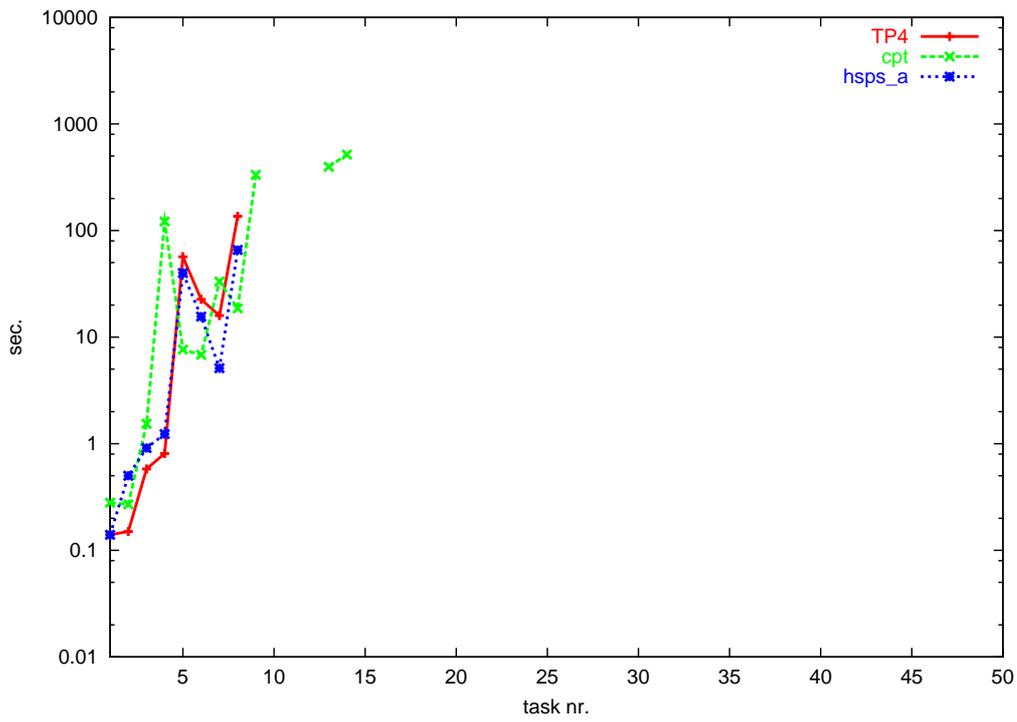


Figura 4.14: *temporal-no-tankage*, tempo de execução, ótimos

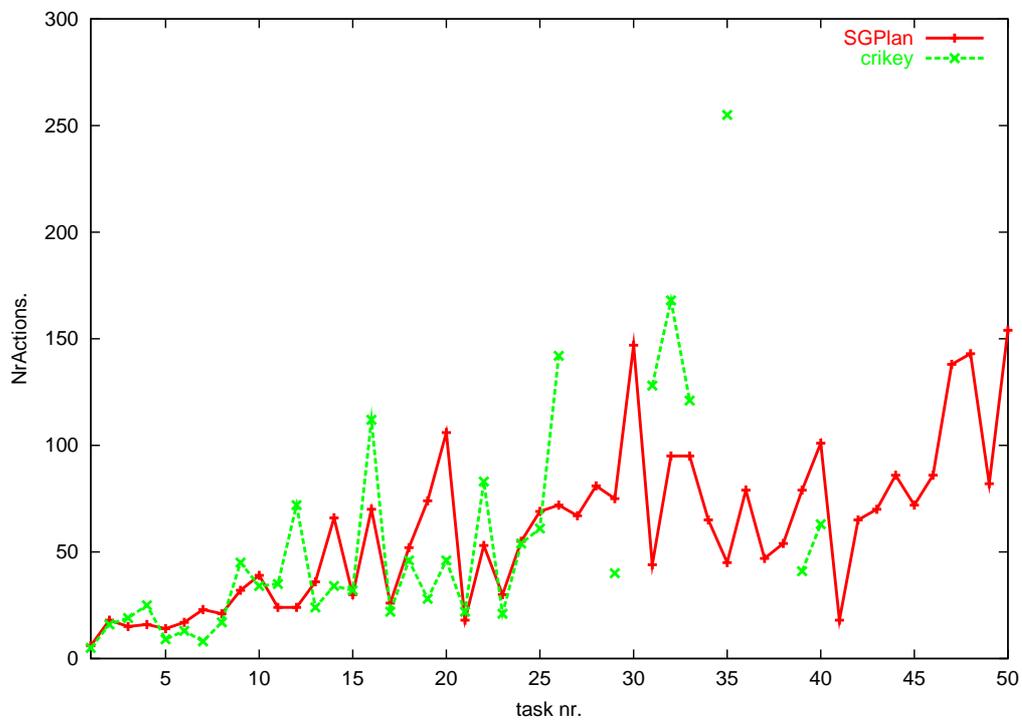


Figura 4.15: *temporal-no-tankage*, número de ações

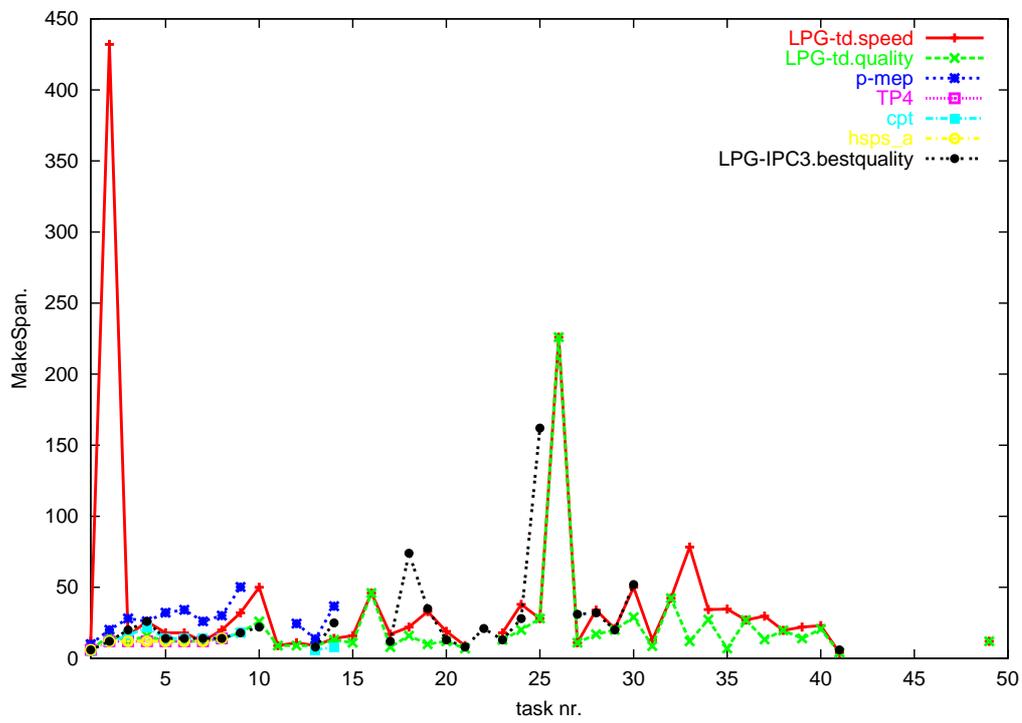


Figura 4.16: *temporal-no-tankage*, makespan

4.3.4 Com tancagem, com tempo

Esta Seção analisa os resultados para o conjunto *temporal-tankage*, descrito na Seção 4.2.4. De uma forma geral a adição do componente temporal nas instâncias com tancagem teve o mesmo efeito da adição deste componente nas instâncias sem tancagem. Desta forma, as observações feitas na Seção anterior se aplicam também a esta Seção.

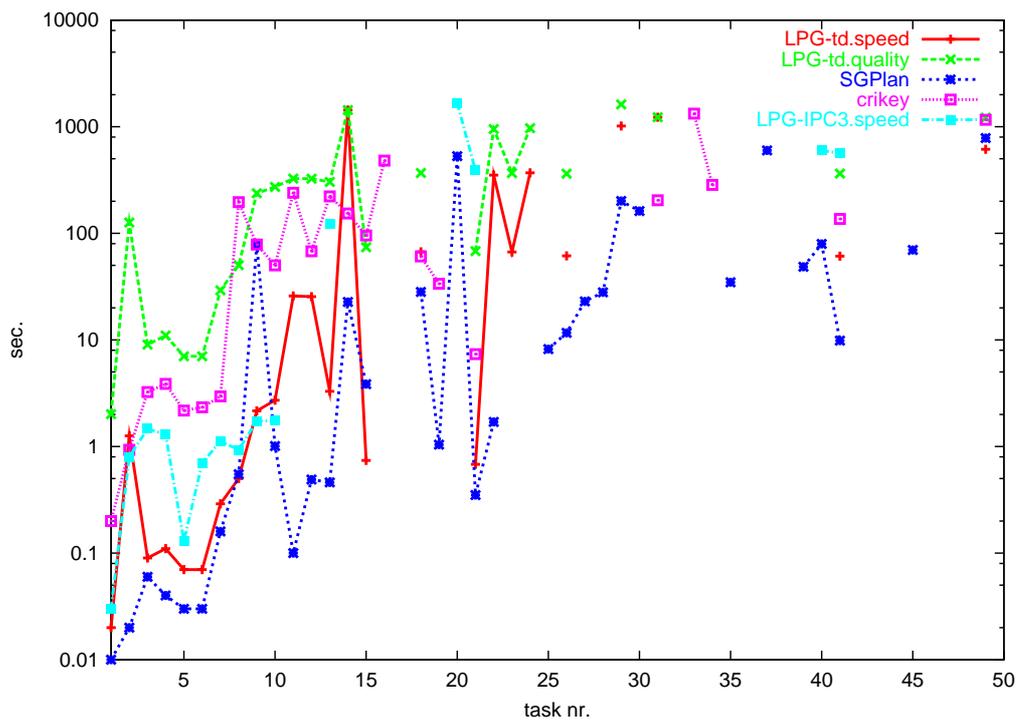


Figura 4.17: *temporal-tankage*, tempo de execução, sub-ótimos

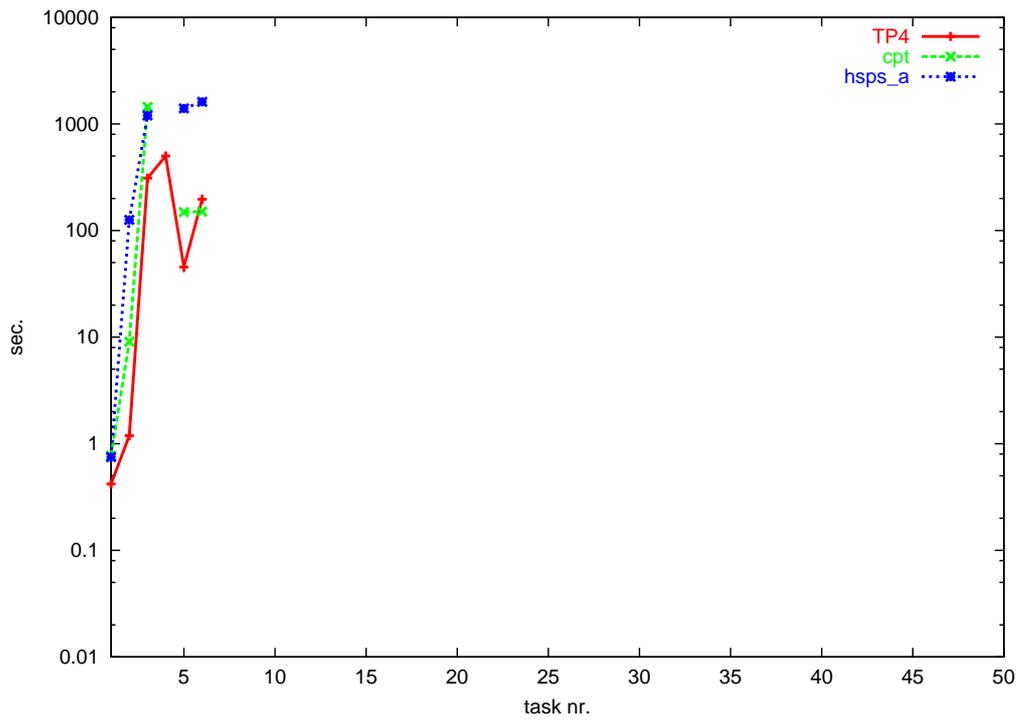


Figura 4.18: *temporal-tankage*, tempo de execução, ótimos

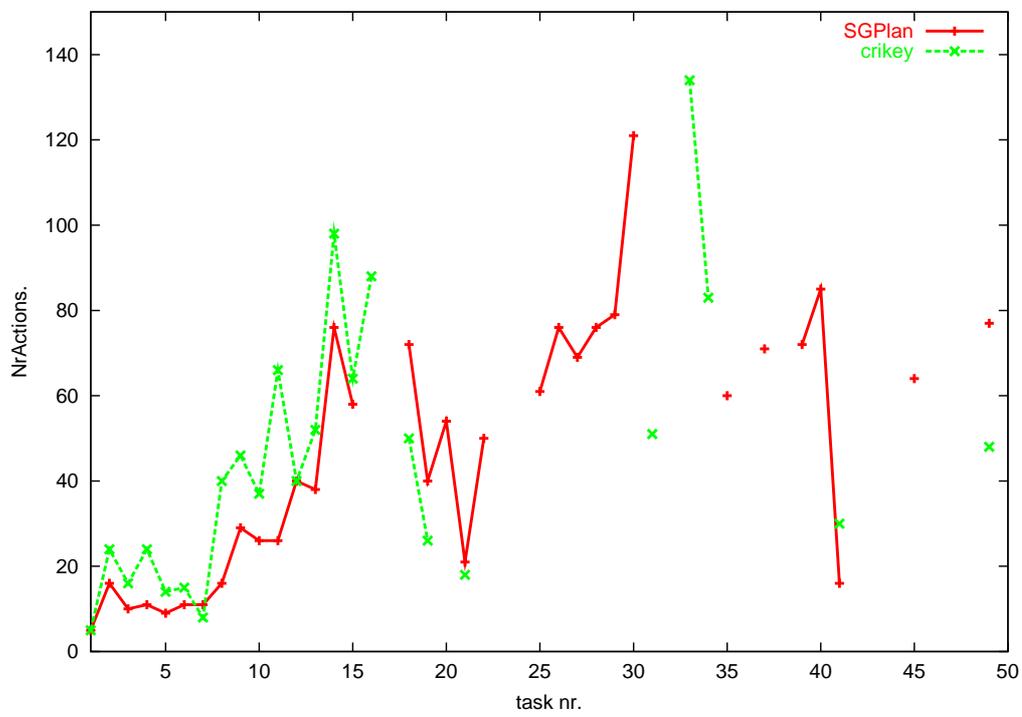


Figura 4.19: *temporal-tankage*, número de ações

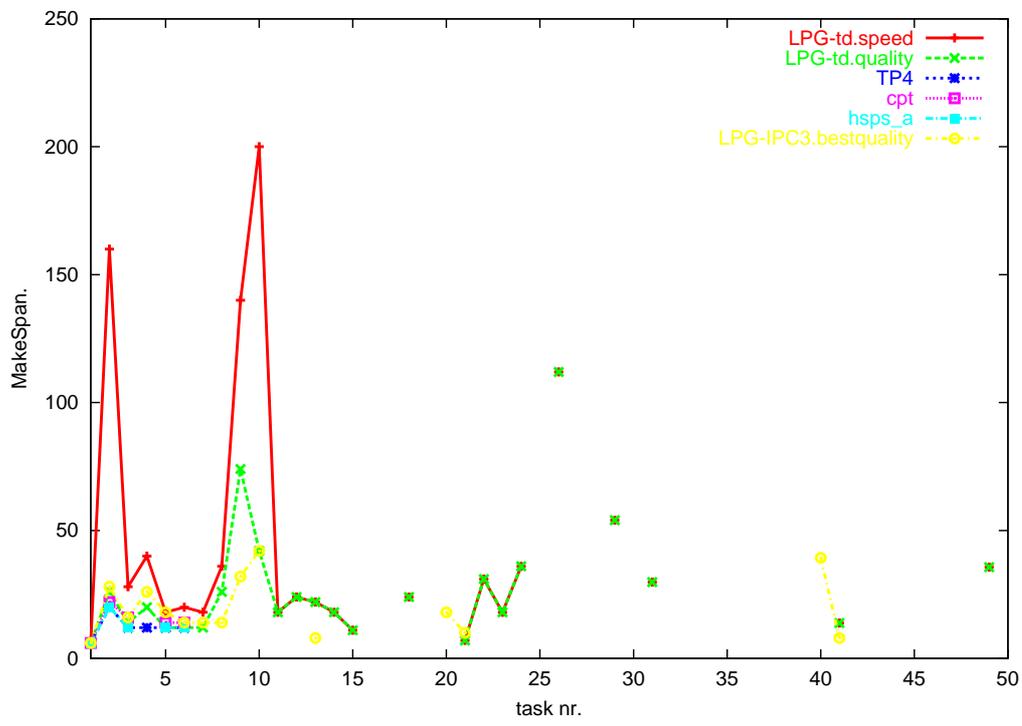


Figura 4.20: *temporal-tankage*, makespan

4.3.5

Sem tancagem, com tempo e prazo

Esta Seção apresenta os resultados para o conjunto *temporal-no-tankage-deadlines*, descrito na Seção 4.2.5. O pequeno número de planejadores que suportam literais temporizados (Seção 3.3.1) prejudica qualquer análise mais detalhada sobre os gráficos das Figuras 4.21 e 4.22.

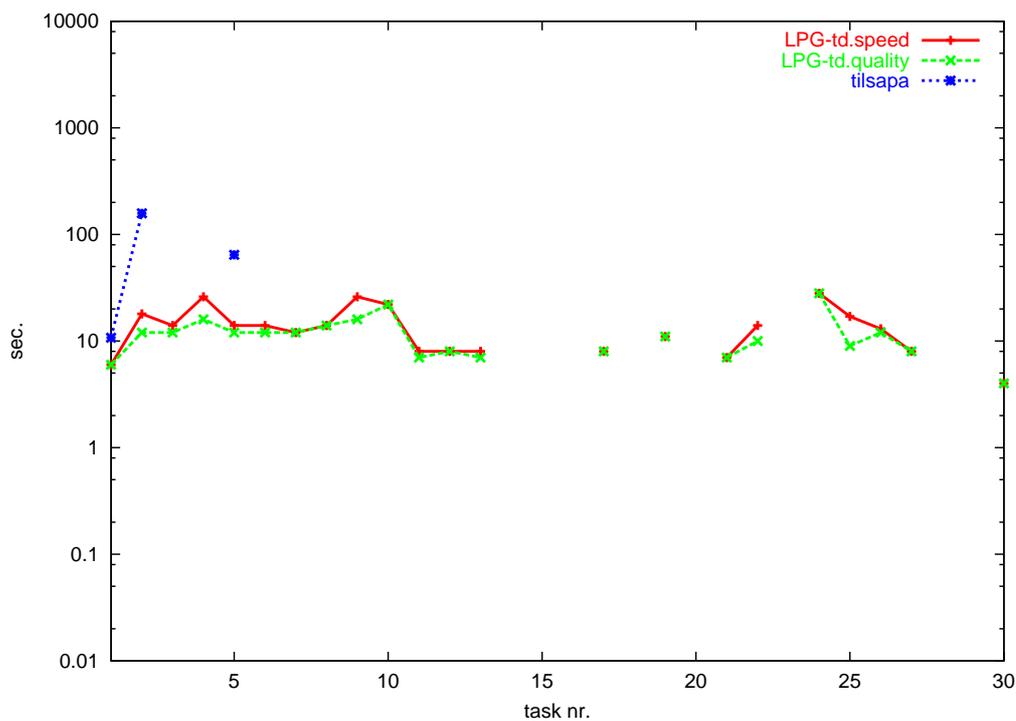


Figura 4.21: *temporal-no-tankage-deadlines*, tempo de execução, sub-ótimos

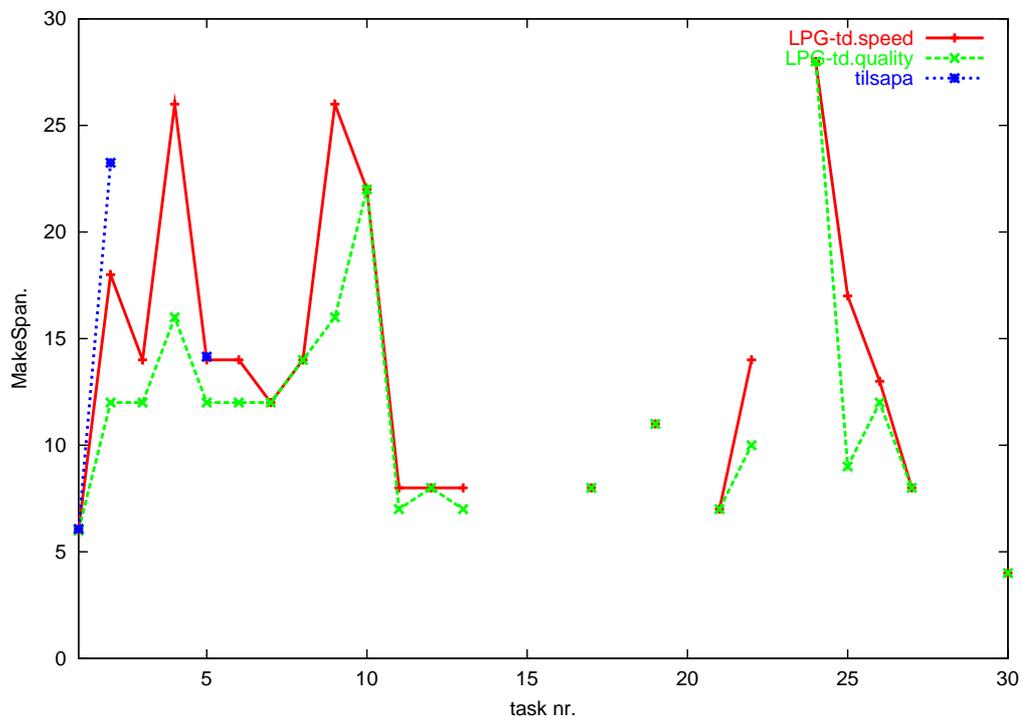


Figura 4.22: *temporal-no-tankage-deadlines*, makespan

4.4 Conclusões

O benchmark do Pipesworld, mesmo sendo uma simplificação do problema real apresentado no capítulo 2 tanto em termos de características quanto em termos de escala, mostrou-se desafiador para o estado da arte dos planejadores de propósito geral. Muitas das instâncias com tancagem e efeitos temporais não foram resolvidas por nenhum planejador no intervalo de 30 minutos fixado pelo comitê organizador da competição.

Outra constatação é o domínio total, em termos de desempenho, de planejadores sub-ótimos e que utilizam busca heurística no processo de planejamento. Todos os planejadores citados na análise dos resultados pertencem a esta categoria. Estes planejadores diferem entre si em dois aspectos principais: na forma como a função heurística é derivada automaticamente e na estratégia de busca utilizada².

O fato de os planejadores não terem tido um desempenho espetacular nas instâncias do Pipesworld não significa que as técnicas utilizadas por eles não possam ser aplicáveis no desenvolvimento de um Planejador Especializado em Dutos, que chamaremos genericamente de PED.

Como, na prática, heurísticas melhores do que as derivadas automaticamente podem ser fornecidas sobre especialistas no domínio, este não parece ser um aspecto crítico no desenvolvimento do PED. O capítulo 6 mostra que, com heurísticas simples e específicas para o Pipesworld, é possível superar um dos planejadores de propósito geral de melhor desempenho, utilizando a mesma estratégia de busca.

A dificuldade neste ponto é que, conforme descrito na Seção 3.1, modelos escritos em PDDL não permitem que informação sobre funções heurísticas sejam fornecidas. Isto impede que o projetista do PED utilize o PDDL para testar as diversas opções de estratégia de busca conhecidas, já que estes testes na verdade avaliam de forma simultânea os dois aspectos do planejador, a estratégia de busca e a forma como a função heurística é derivada automaticamente.

Outra dificuldade do uso de PDDL no projeto do PED é a pequena expressividade da linguagem, já antecipadas na Seção 3.4.1. Muitas destas limitações podem ser resolvidas através da mudança do modelo utilizado para representar o sistema que está sendo analisado da representação em PDDL para uma forma mais genérica de simuladores a eventos discretos.

²A Seção 6.1 apresenta um pequeno exemplo da abordagem utilizada por estes planejadores.

Uma análise mais detalhada das estratégias de busca dos planejadores citados neste capítulo, incluindo a justificativa para os planos longos gerados pelo YAHSP e FAP é mostrada na Seção 6.1.3.

O PLANSIM, apresentado do capítulo 7, procura endereçar estes pontos, definindo um arcabouço para a integração de funções heurísticas, simuladores a eventos discretos e estratégias de busca.