TÊMPERA SIMULADA APLICADA AO PROBLEMA DE ESCALONAMENTO COM RESTRIÇÃO DE RECURSOS

Luciano Lessa Lorenzoni *#
FAESA - UFES
luciano@faesa.br

Carlos Albuquerque *
FAESA
camvra@bol.com.br

Rafael Dias Nascimento *
FAESA
rafdias@gazetaonline.com.br

* FAESA - Ciência da Computação
Rua Anselmo Serrat, 199, Ilha de Mote Belo, Vitória - ES, CEP 29040-410

UFES - Departamento de Informática
Av Fernando Ferrari, S/N, Goiabeiras, Vitória - ES, CEP 29060-900

Resumo

Neste artigo apresentamos uma implementação da meta-heurística têmpera simulada aplicada ao problema de escalonamento com restrição de recursos com múltiplos modos de processamento. O problema consiste em identificar o modo e o tempo de início de processamento de cada tarefa e tem como objetivo minimizar o maior tempo de finalização dentre todas as tarefas, de forma que as restrições de recurso e de precedência sejam satisfeitas. Apresentamos ainda uma nova forma de representação para o escalonamento denominada representação em árvore com chave randômica que foi utilizada na implementação do algoritmo. O algoritmo foi testado utilizando instâncias padrões do problema geradas pelo ProGen. Os resultados computacionais obtidos indicaram um bom desempenho do algoritmo.

Palavras-chave: Problema de escalonamento com restrição de recursos, Múltiplos modos, Têmpera simulada.

Abstract

In this paper we present an algorithm based on simulated annealing procedures for solving the multi-mode resource constrained project scheduling problem. The problem consists of identifying a mode for each task and a starting time for its processing respecting precedence and resource constraints with the objective of minimizing the makespan of the project. We present further a new way to represent the scheduling called Random Key-Tree Representation that was applied in the implementation of algorithm. The algorithm was tested with several instances generated by standard project generator ProGen. The Computational results have shown a good algorithm performance.

Keywords: Resource constrained scheduling problem, Multiple Modes, Simulated annealing.

1 Introdução

O problema de escalonamento com restrição de recursos está associado à alocação de recursos escassos a um conjunto de tarefas num dado tempo. Escalonar significa determinar o tempo de início de execução de cada uma das tarefas e os recursos alocados para a sua execução até que todas tenham sido executadas sob as restrições impostas com o objetivo de minimizar o comprimento do escalonamento (tempo de processamento das tarefas) ou um outro critério estabelecido no contexto do problema (Blazewicz et al. 1996). Aplicações desse problema podem ser encontradas em diversas situações do mundo real, por exemplo, no planejamento da produção, no genrenciamento de projetos e no seqüenciamento de tarefas pela unidade central do computador.

A teoria do escalonamento é caracterizada por uma grande variedade de tipos de problemas. Esses problemas são classificados de acordo com os tipos de recursos existentes, das características das tarefas a serem executadas e do objetivo a ser alcançado. Uma descrição dos principais modelos pode ser encontrada Pinedo (1995), Blazewicz et al. (1996) e Brucker (1998).

A resolução desses problemas tem sido um desafio dada a sua complexidade. Em geral, os problemas de escalonamento são considerados NP-Difíceis e as abordagens utilizadas para a resolução dos diversos modelos têm sido elaboradas através de métodos de relaxação, algoritmos enumerativos e algoritmos de aproximação.

Especificamente, neste trabalho, o foco é a classe de problemas de escalonamento com restrição de recursos com múltiplos modos de processamento. Dentre os algoritmos exatos desenvolvidos para a resolução desse problema destacamos a abordagem *branch-and-bound* desenvolvida por Sprecher et al. (1997) e a proposta por Sprecher e Drexl (1998) baseada em árvore de precedência. Ressaltamos também os algoritmos heurísticos desenvolvidos por Drexl (1991) e e Drexl & Grünewald (1993) que analisaram regras de prioridade baseadas em heurísticas multi-passo, Özdamar (1996) e Hartmann (1997) que desenvolveram uma versão para o algoritmo genético, Verhoeven (1998) para a busca tabu e Lorenzoni et al. (2001) para a colônia de formigas. Já Kolisch & Drexl (1997) desenvolveram algoritmos de busca local próprios.

Neste artigo propomos a aplicação da meta-heurística têmpera simulada para a resolução do problema, utilizando uma nova forma de representação do escalonamento denominada representação em árvore com chave randômica. O artigo está estruturado da seguinte forma. Na seção 2 caracterizamos o problema de escalonamento com restrição de recursos com múltiplos modos de processamento e na seção 3 apresentamos a nova forma de representação para o escalonamento. O algoritmo baseado na têmpera simulada implementada para a resolução do problema é descrito na seção 4. Os resultados computacionais para 15 instâncias padrões encontradas na literatura são apresentados na seção 5 e, finalmente na seção 6, as conclusões e perspectivas futuras.

2 Escalonamento com restrição de recursos com múltiplos modos de processamento

No problema clássico de escalonamento com restrição de recursos com múltiplos modos de processamento, um conjunto de tarefas tem que ser escalonado com o objetivo de minimizar, dentre todas as tarefas, o maior tempo de finalização de execução das tarefas, de tal forma que satisfaça as relações de precedências entre as tarefas e as restrições de capacidade de todos os recursos.

Dessa forma, uma instância do problema de escalonamento com restrição de recursos com múltiplos modos de processamento consiste de um conjunto de tarefas T, um conjunto de Recursos $R = R^{ren} \quad R^{nren}$, onde R^{ren} contém os recursos renováveis e R^{nren} os não renováveis. Uma quantidade constante c_r do recurso $r \quad R$ está disponível. Caso o recurso r seja renovável, essa quantidade está disponível em qualquer momento, e, caso seja não renovável, essa quantidade está disponível para toda a execução do escalonamento.

Para cada tarefa $j \quad T$, um conjunto M_j indica os diversos modos de processamento de j. Como cada tarefa pode ser processada de diversos modos, o tempo de processamento da tarefa num recurso particular depende do modo de processamento escolhido, assim $p_{j,r,m}$ corresponde ao tempo de processamento da tarefa j, utilizando o recurso r, caso seja executada no modo de processamento m M_j e $q_{j,r,m}$ indica a quantidade de recurso r, utilizada para processar j, no modo m.

Uma relação binária $\prec$ define uma ordem parcial em T, indicando as precedências entre as tarefas.

Assumiremos o modelo não preemptivo, ou seja, uma vez que a execução da tarefa tenha sido

A representação do escalonamento utilizada neste trabalho é uma nova forma de representação, denominada representação em árvore com chave randômica (Random Key-Tree Representation), que será descrita a seguir.

Podemos dizer que o escalonamento de n tarefas consiste de $k = 1, 2, ..., n$ etapas. Seja E_k o conjunto das tarefas, denominadas elegíveis, que podem ser processadas na etapa k, ou seja temos em E_k as tarefas cujas as predecessoras já foram todas escalonadas. Em cada etapa k do escalonamento uma tarefa em E_k é selecionada. A seguir, determina-se o modo de execução dessa tarefa e processa-a. Associamos então, a cada etapa k, um nível k de uma árvore de decisão cujos nós são as tarefas em E_k.

Na representação em árvore com chave randômica o escalonamento com restrição de recursos com múltiplos modos de processamento é representado por uma matriz

$$A \quad \begin{matrix} \vec{} \\ \vec{} \end{matrix}$$

onde os vetores $\vec{}$ ($_1$, $_2$, ..., $_n$) e $\vec{}$ ($_1$, $_2$, ..., $_n$) associam dois valores $_k$, $_k$ [0,1] a cada nível k da árvore, que indicarão, respectivamente, a tarefa e o seu modo de processamento na etapa k do escalonamento.

3.1 Procedimento decodificar

Para cada uma das tarefas pertencentes ao nível k da árvore associa-se uma faixa de valores da seguinte forma: o intervalo $0, \frac{1}{|E_k|}$ à primeira tarefa do conjunto E_k, o intervalo $\frac{1}{|E_k|}, \frac{2}{|E_k|}$ à segunda tarefa, e assim sucessivamente, até à última tarefa o intervalo $\frac{|E_k| \quad 1}{|E_k|}, 1$. Também associa-se uma faixa de valores para cada modo de processamento, de cada uma das tarefas, da seguinte forma: ao primeiro modo de processamento o intervalo $0, \frac{1}{|M_j|}$, ao segundo modo de processamento o intervalo $\frac{1}{|M_j|}, \frac{2}{|M_j|}$, e assim sucessivamente, até ao último modo de processamento o intervalo $\frac{|M_j| \quad 1}{|M_j|}, 1$.

Para se obter o escalonamento a partir de A selecione em cada etapa k, a tarefa j E_k tal que $_k$ pertença à faixa correspondente associada à tarefa j. Uma vez que, a tarefa j foi selecionada, selecione o modo m M_j de processamento tal que $_k$ pertença a faixa correspondente associada ao modo de processamento m.

Para ilustração do procedimento, considere o seguinte problema de escalonamento com restrição de recursos com múltiplos modos de processamento. Sejam 5 tarefas e 3 recursos renováveis cujas capacidades são respectivamente 3, 4 e 3 unidades. Os arcos ligando as tarefas, na Figura 1, indicam as relações de precedência. As tarefas 0 e 6 são artificiais. Na Tabela 1 temos os possíveis modos de processamento de cada uma das tarefas.

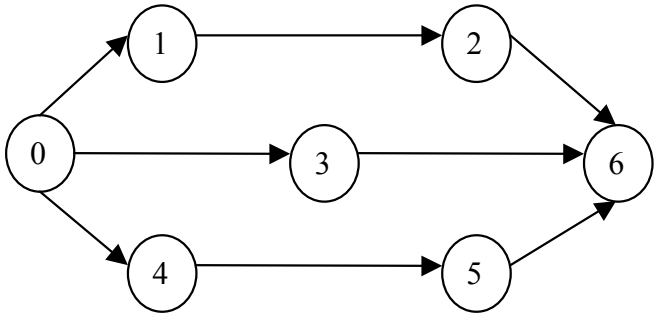

Figura 1: Relações de precedência

Tarefa	*Modo*	*Duração*	R_1	R_2	R_3
1	1	3	0	2	1
	2	2	2	2	0
	3	1	3	0	1
2	1	4	1	1	0
	2	2	2	2	0
3	1	1	0	3	2
	2	3	2	0	1
4	1	2	3	2	0
	2	2	1	0	2
	3	3	0	1	2
5	1	3	1	3	0
	2	4	1	0	1
	3	5	3	0	3
	4	2	4	2	0

Tabela 1: Modos de processamento

Tomando a matriz

$$A \quad \begin{matrix} 0.37 & 0.53 & 0.18 & 0.76 & 0.34 \\ 0.72 & 0.62 & 0.42 & 0.17 & 0.18 \end{matrix}$$

como a representação do escalonamento temos na Figura 2 a árvore de decisão gerada e na Tabela 2 temos, para cada nível da árvore, os intervalos associados a cada uma das tarefas pertencentes ao nível, bem como, os intervalos associados a cada modo de processamento de cada uma das tarefas em E_k.

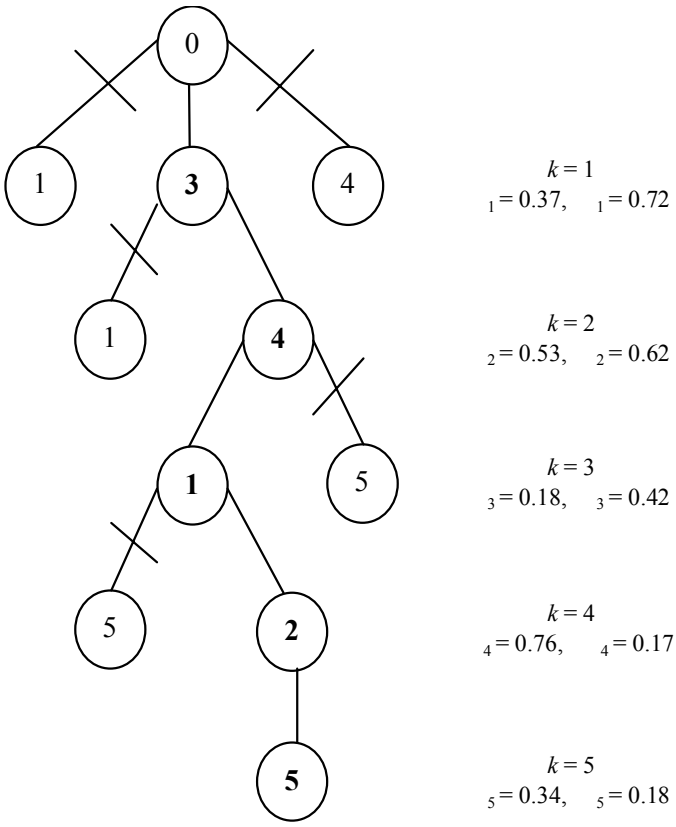

Figura 2: Árvore de precedência

k	E_k	tarefa → intervalo	(tarefa,modo) → intervalo
1	$E_1 = \{1,3,4\}$	1 → [0,0.33) 3 → [0.33,0.66) 4 → [0.66,1]	(1,1) → [0,0.33) (1,2) → [0.33,0.66) (1,3) → [0.66,1] (3,1) → [0,0.5) (3,2) → [0.5,1] (4,1) → [0,0.33) (4,2) → [0.33,0.66) (4,3) → [0.66,1]
2	$E_2 = \{1,4\}$	1 → [0,0.5) 4 → [0.5,1]	(1,1) → [0,0.33) (1,2) → [0.33,0.66) (1,3) → [0.66,1] (4,1) → [0,0.33) (4,2) → [0.33,0.66) (4,3) → [0.66,1]
3	$E_3 = \{1,5\}$	1 → [0,0.5) 5 → [0.5,1]	(1,1) → [0,0.33) (1,2) → [0.33,0.66) (1,3) → [0.66,1] (5,1) → [0,0.25) (5,2) → [0.25,0.5) (5,3) → [0.5,0.75) (5,4) → [0.75,1]
4	$E_4 = \{5,2\}$	5 → [0,0.5) 2 → [0.5,1]	(5,1) → [0,0.25) (5,2) → [0.25,0.5) (5,3) → [0.5,0.75) (5,4) → [0.75,1] (2,1) → [0,0.5) (2,2) → [0.5,1]
5	$E_5 = \{5\}$	5 → [0,1]	(5,1) → [0,0.25) (5,2) → [0.25,0.5) (5,3) → [0.5,0.75) (5,4) → [0.75,1]

Tabela 2: Intervalos associados

Como $_1 = 0.37$ [0.33,0.66), logo a tarefa 3 será a selecionada na etapa $k = 1$ e o modo de processamento será o modo 2 pois $_1 = 0.72$ [0.5,1]. De modo análogo temos que na etapa $k = 2$ a tarefa selecionada será a 4 com o modo de processamento 2, na etapa $k = 3$ a tarefa selecionada será a 1 com o modo de processamento 2, na etapa $k = 4$ a tarefa selecionada será a 2 com o modo de processamento 1 e na etapa $k = 5$ a tarefa selecionada será a 5 com o modo de processamento 1. Na Figura 3 temos o escalonamento associado a representação A.

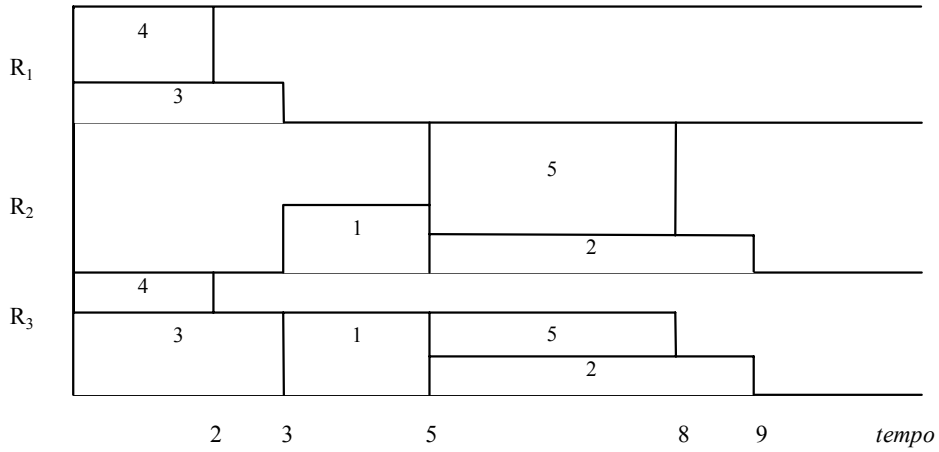

Figura 3: Escalonamento associado a *A*

4 Têmpera simulada aplicada ao problema de escalonamento

A têmpera simulada, uma tradução para *simulated annealing*, é uma abordagem heurística desenvolvida para encontrar uma boa solução, não necessariamente ótima, para problemas de otimização, dentro de um tempo computacional razoável. Ela tem sido aplicada com sucesso tanto a problemas de otimização combinatória (discreta) como a problemas de otimização global (contínua).

Essa abordagem foi proposta independentemente por Kirkpatrick et al. (1983) e Cerny (1985) e está baseada na simulação algorítmica do processo físico de têmpera dos materiais. Segundo Viana (1998), o processo de têmpera consiste em submeter os materiais inicialmente a elevadas temperaturas e reduzi-las gradualmente até atingirem, com aumentos e reduções do estado de energia, o equilíbrio térmico, tornando-os rígidos e consistentes. O resfriamento deve ser feito de forma cuidadosa (lentamente) para que o material atinja um estado fundamental (baixa energia), de modo a produzir um material com uma estrutura que tenha a resistência desejada. Caso o resfriamento não seja feito de forma adequada, o sistema não terá tempo suficiente para alcançar o equilíbrio térmico, podendo o estado final ter alta energia e como conseqüência o material apresentar defeitos em sua estrutura, não alcançando as características desejadas.

Na Figura 4 temos ilustrada a analogia entre o processo físico de têmpera dos materiais e os problemas de otimização.

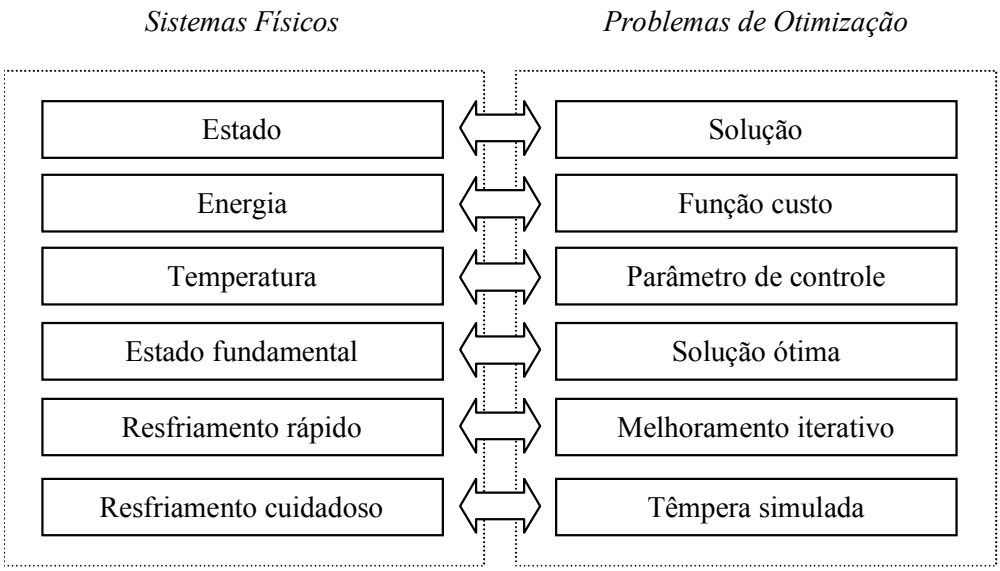

Figura 4: Analogia entre sistemas físicos e problemas de otimização

Nessa analogia, os diferentes estados dos materiais correspondem às diferentes soluções dos problemas, sendo que o estado fundamental está a

equilíbrio térmico novas soluções vizinhas S´ são geradas a partir de S através de uma perturbação em S. No nosso caso, como as soluções são codificadas a partir da representação em árvore com chave randômica, conforme descrição na seção 3, novas soluções são obtidas selecionando-se aleatoriamente um nível *k* da árvore de decisão e escolhendo randomicamente dois novos valores $_k$ e $_k$ para formarem a nova solução S´.

Um novo estado S´ é aceito como uma nova solução se tiver um melhor custo que o estado S (= C(S´) – C(S) 0) ou, se mesmo tendo um custo pior, o valor exp(- /T), for maior que um número gerado aleatoriamente entre zero e um. A probabilidade de aceitar uma mudança deste tipo é chamada critério de Metropolis, estabelecido a partir do fator de Boltzmann, que é dado pela função exp(- /T). Analisando a expressão exp(- /T) podemos observar que quanto maior o valor de T maior a possibilidade de soluções com um custo pior serem aceitas. À medida que T se aproxima de zero mais mudanças indesejáveis são rejeitadas. No início do processo é necessário a temperatura relativamente alta para evitar o bloqueio nas configurações com ótimo local.

Em geral, considera-se que o equilíbrio térmico foi atingido quando um certo número de configurações L forem aceitas ou nenhuma melhoria ocorreu após P tentativas. Neste trabalho o plano de resfriamento foi obtido a partir da expressão T = T com um valor adequado para [0.8,0.99) conforme proposto por Romeo et al. (1984).

Uma descrição mais detalhada da meta-heurística têmpera simulada, bem como a determinação dos diversos parâmetros, pode ser encontrada em Eglese (1990) e em Viana (1998).

O algoritmo têmpera simulada básico foi estendido com o intuito de melhorar o seu desempenho através das rotinas de pré-processamento e busca local descritas a seguir.

4.1 Pré-Processamento

A rotina de pré-processamento foi introduzida por Sprecher et al. (1997) com o objetivo de reduzir o espaço de busca, diminuindo o número de soluções viáveis e inviáveis, sem contudo afetar o conjunto de soluções ótimas, através da eliminação dos modos não executáveis, ineficientes e dos recursos não renováveis redundantes.

Segundo Sprecher et al. (1997) um modo é dito não executável se a sua execução viola as restrições de recursos renováveis ou não renováveis em qualquer escalonamento e é considerado ineficente se a sua duração não é menor e a sua requisição de recursos não é menor que os outros modos de processamento da mesma tarefa. Um recurso não renovável é chamado redundante se o somatório da maior requisição desse recurso por cada tarefa não excede a capacidade do recurso.

Observando a Tabela 1 verificamos que o modo de execução 4 da tarefa 5 é não executável, pois a requisição de 4 unidades de R_1 para o processamento da tarefa ultrapassa a capacidade de R_1 que é de 3 unidades por período. Já o modo de execução 3 da tarefa 5 é dito ineficiente, já que, requisita os mesmos recursos, R_1 e R_3, que o modo de execução 2, só que em maiores quantidades e com maior tempo de processamento.

4.2 Busca local

O método de busca local implementado é baseado na definição *multi-mode left shift*. Um *multi-mode left shift* de uma tarefa *j* é uma ação executada sobre o escalonamento que reduz o tempo de finalização da tarefa *j* sem modificar os modos de processamento e os tempos de finalizações das outras tarefas e sem violar as restrições de recursos, mas o modo da tarefa *j* pode ser mudado.

A definição *multi-mode left shif*, assim como, a rotina de pré-processamento foi introduzida por Sprecher et al. (1997) com o objetivo de acelerar o algoritmo branch-and-bound desenvolvido para a resolução do problema de escalonamento com restrição de recursos com múltiplos modos de processamento. Essa definição também foi utilizada com sucesso por Hartmann (1997) no desenvolvimento da versão do algoritmo genético para o mesmo problema.

A tentativa de melhorar um escalonamento viável a partir de uma busca local é feita da seguinte forma: para cada tarefa do escalonamento, seguindo a ordem de processamento, o primeiro *multi-mode left shift*, se encontrado, é aplicado ao escalonamento. O resultado é um escalonamento com o maior

tempo de finalização, dentre todas as tarefas, igual ou menor ao original. Esse método é aplicado a todas as soluções viáveis geradas pelo algoritmo têmpera simulada.

5 Experimentos Computacionais

Um conjunto de problemas testes construídos pelo gerador ProGen, desenvolvido por Kolisch & Sprecher (1997), foi utilizado para testar o desempenho do algoritmo implementado. Os problemas estão disponíveis na biblioteca denominada PSPLIB *(project scheduling problem library)*. Para compor a massa de testes foram escolhidas aleatoriamente, 5 instâncias com 10 tarefas, 3 instâncias com 16 tarefas e 7 instâncias com 20 tarefas. Nessas instâncias existem dois recursos renováveis e dois não renováveis e cada uma das tarefas pode ser executada em três diferentes modos.

O algoritmo desenvolvido neste trabalho foi desenvolvido em Visual C++ e executado sobre a plataforma Windows NT num computador pessoal Pentium III 1GHz. A Tabela 3 exibe os resultados obtidos com a seguinte escolha dos valores dos parâmetros, = 0.8, P = 100*|*T*|, L = 10*|*T*|, onde |*T*| é o número de tarefas da instância do problema. A temperatura inicial T_0 foi calculada a partir da expressão - $C^+/\ln(0.8)$ proposta por Johnson et al. (1987), onde C^+ é a média aritmética, para um número randômico de perturbações, dos incrementos da função objetivo. Nessa tabela, as colunas TF_{PSPLIB} e TF_{TS} indicam a solução ótima e os tempos de finalização apresentados pelo algoritmo têmpera simulada, respectivamente. Nas colunas d_a e d_r, têm-se a diferença em valor absoluto e valor relativo, respectivamente, entre as soluções apresentadas e t(s) fornece o tempo médio, em segundos, de execução do algoritmo têmpera simulada.

Instância	|*T*|	TF_{PSPLIB}	TF_{TS}	d_a	d_r	t(s)
J102_2	10	20	20	0	0	0.32
J104_3	10	23	23	0	0	0.17
J1028_8	10	16	26	0	0	0.12
J1048_5	10	14	14	0	0	0.17
J1051_5	10	29	29	0	0	0.16
J1620_3	16	28	28	0	0	0.35
J1642_2	16	29	30	1	3.4	3.3
J1653_8	16	25	25	0	0	1.37
J2011_10	20	26	26	0	0	1.17
J2025_5	20	30	31	1	3.3	1.46
J2045_6	20	36	38	2	5.5	6.6
J2051_5	20	28	28	0	0	2.42
J2051_6	20	22	23	1	4.5	1.44
J2062_10	20	26	28	2	7.7	1.38
J2063_7	20	36	36	0	0	3.26

Tabela 3 : Resultados para o problema de escalonamento com restrição de recursos com múltiplos modos de processamento

De acordo com a tabela, observamos que o algoritmo atingiu para 10 das 15 instâncias testadas a solução ótima. Sendo que, para as 5 instâncias que não atingiram a solução ótima, a média do desvio, em valores relativos, foi de 4.9%, e em termos absolutos, foi de 1.4. Para as instâncias testadas podemos considerar que o algoritmo teve um bom desempenho.

6 Conclusões

Neste artigo foi apresentada uma implementação da abordagem têmpera simulada para a resolução de problemas de escalonamento com restrição de recursos com múltiplos modos de processamento. Neste algoritmo utilizou-se para caracterizar as soluções, uma nova forma de representação do escalonamento, denominada representação em árvore com chave randômica. O algoritmo testado em para 15 instâncias padrões apresentou bons resultados, o que nos encoraja

aperfeiçoar o projeto do algoritmo através da exploração do conhecimento específico do domínio do problema, visto que, para outros problemas clássicos de escalonamento, como o *job shop*, a utilização desse conhecimento tem sido útil no desenvolvimento de boas estratégias de resolução do problema conforme Nowick & Smutnicki (1996).

Uma outra direção para dar continuidade ao trabalho seria estender o algoritmo a aplicações reais. Uma dessas aplicações, que vem sendo estudada, encontra-se no contexto portuário, mais especificamente no gerenciamento de filas de navios, no qual as janelas de tempo para a chegada dos navios no porto seriam obtidas como solução de um problema de escalonamento com restrição de recursos com múltiplos modos de processamento.

7 Referências

Blazewicz, J.J., Ecker, K. H., Pesch, E., Schmidt, G., Weglarz, J., 1996. Scheduling Computer and Manufacturing Processes. Springer Verlag.

Brucker, P., 1998. Scheduling Algorithms, Springer, Berlin.

Bullnheimer, B., Hartl, R. F., Strauss, C., 1998. Applying the ant system to the vehicle routing problem. In I. H. Osman S. Voss, S. Martello and C. Roucariol, editors, Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization, Kluwer Academics, 109-120.

Cerny, V., 1985. Thermodynamical approach toto the traveling salesman problem: An efficient simulation algorithm, Journal of Optimisation Theory and Applications 45, 41-51.

Drexl, A., 1991. Scheduling of project networks by job assignment, Management Science, 37, 1590-1602.

Drexl, A., Grünewald, J., 1993. Nonpreemptive multi-mode resource-constrained project scheduling, IIE Transactions, 25(5), 74-81.

Eglese, RW., 1990. Simulated Annealing: A tool for Operational Reserach. European Journal of Operational Research 46, 276-281.

Johnson, D.S., 1974. Approximation algorithms for combinatorial problemas. J. Computing System Science 9, 256-278.

Hartmann, S., 1997. Project scheduling with multiple modes: A genetic algorithm, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, 435.

Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P., 1983. Optimization by simulated annealing, Science 220, 671-680.

Kolisch, R., Drexl, A., 1997. Local search for nonpreemptive multi-mode resource-constrained project scheduling, IIE Transactions, 29, 987-999.

Kolisch, R., Hartmann, S., 1998. Heuristics algorithms for solving the resource constarined project scheduling problem: classification and compuatational analysis, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, Nº 469.

Kolisch, R., Sprecher, A., 1997. PSPLIB – A project scheduling library. European Journal of Operational Research 112, 205-216.

Lorenzoni, L.L., Ahonen, H., Alvarenga, A. G., 2001. Colônia de Formigas para problemas de escalonamento com restrição de recursos, Anais do XXXIII SBPO – Simpósio Brasileiro de Pesquisa Operacional, 1001-1010.

Nowicki, E., Smutnicki, C., 1996. A fast taboo search algorithm for the job shop problem, Mangement Science, 42(6), 797-813.

Özdamar, L., 1996. A genetic algorithm approach to a general category project scheduling problem, Research Report, Marmata University, Istanbul.

Pinedo, M., 1995. Scheduling – theory, algorithms and systems, Prentice-Hall, Englewood Cliffs, NJ.

Romeo, F., Sangiovanni-Vicentelli, A. L., Sechen, C., 1984. Research on simulated annealing at Berkeley, Proceeding IEEE International Conference on Computer Design, 652-657, Post Chester, November.

Schoonderwoerd, R., Holland, O., Bruten, J., Rothkrantz, L., 1996. Ant-based load balancing in telecommunications networks. Adaptative Behavior, 5(2), 169-207.

Slowinski, R., Soniewicki, B., Weglarz, J., 1994. DSS for multi-objective project scheduling subject to multiple-category resource constraints, European Journal of Operational Research 79, 220-229.

Sprecher, A., Hartmann, S., Drexl A., 1997. An exact algorithm for project scheduling with multiples modes. OR Spektrum, 19, 195-203.

Sprecher, A., Drexl A., 1998. Solving multi-mode resource constrained project scheduling problems by a simple, general and powerful sequencing algorithm. European Journal of Operational Research 107, 431-450.

Stützle, T., Hoos, H., 1997. Improvements on the ant system: Introducing MAX-MIN ant system. In Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms, 245-249. Springer Verlag, Wien.

Verhoeven, M. G. A., 1998. Tabu search for resource-constrained scheduling. European Journal of Operational Research, 106, 266-276.

Viana, V., 1998. Meta-Heurísticas e programação paralela em otimização combinatória, Edições UFC, Fortaleza, Ceará.