



GRUPO VII

GRUPO DE ESTUDO DE PLANEJAMENTO DE SISTEMAS ELÉTRICOS (GPL)

TIME ASSÍNCRONO INICIALIZADOR DE MÉTODOS COMBINATORIAIS PARA
PLANEJAMENTO DA EXPANSÃO DA TRANSMISSÃO

Sérgio Azevedo de Oliveira* Cláudio R. Thomas de Almeida Alcir Monticelli

UNESP-FEIS-DEE

UNICAMP-FEEC-DSEE

RESUMO

Este artigo trata da aplicação de *Times Assíncronos (Times-A)* ao planejamento da expansão de sistemas de transmissão de energia elétrica. Uma parte importante desse processo é a determinação de famílias de soluções iniciais contendo características atrativas, isto é, configurações com linhas e transformadores e conjuntos de linhas e transformadores (blocos construtivos) que podem formar parte das soluções ótimas, ou quase ótimas, obtidas via métodos como os Algoritmos Genéticos e Busca Tabu. *Times-A* foram desenvolvidos para gerar estas configurações iniciais. Esses times utilizam diferentes métodos heurísticos construtivos disponíveis na literatura, os quais são distribuídos em uma rede heterogênea de computadores e executados de modo concorrente. Testes foram feitos para sistemas de pequeno, médio e grande porte.

PALAVRAS-CHAVE

Times Assíncronos; Processamento Paralelo e Distribuído; Métodos Heurísticos Construtivos; Planejamento da Expansão da Transmissão.

1.0 INTRODUÇÃO

O planejamento da expansão de redes de transmissão de sistemas de energia elétrica determina *quando, onde e que* tipos de linhas e/ou transformadores devem ser instalados na rede a fim de que o sistema opere adequadamente, para uma demanda futura predeterminada, com o menor investimento possível. O planejamento *dinâmico (quando)* em geral é decomposto em subproblemas *estáticos* que tratam das

questões *onde e que tipo* (planejamento em um estágio, de um ano inicial a um ano final, dados). Quando o sistema elétrico é modelado pelo fluxo de carga DC, o problema de planejamento estático é um problema de programação não linear inteira mista (PNLIM). Dadas as dimensões que o problema assume para casos práticos em geral observa-se o fenômeno da explosão combinatória (problema NP-completo). Para uma alternativa de investimento, o problema se reduz a um problema de programação linear cujo objetivo é verificar a factibilidade da alternativa. Soluções deste problema são obtidas via métodos combinatoriais, tais como Algoritmos Genéticos, Busca Tabu, entre outros. Sendo que uma parte importante desse processo é a determinação de famílias de soluções iniciais contendo características atrativas, isto é, configurações com linhas e transformadores e conjuntos de linhas e transformadores (blocos construtivos) que podem formar parte das soluções ótimas, ou quase ótimas.

Anteriormente, em uma época na qual os recursos computacionais eram mais limitados, foram desenvolvidos métodos heurísticos como por exemplo os métodos de Garver, Mínimo Esforço e Mínimo Corte de Carga. Essas metodologias ainda são utilizadas por concessionárias como parte de procedimentos interativos que exigem uma participação ativa dos planejadores. E atualmente, as concessionárias têm em seu parque computacional inúmeras estações de trabalho que podem ser utilizadas, com o auxílio de bibliotecas para processamento paralelo, como uma máquina paralela virtual, onde é possível o processamento concorrente de diversos programas computacionais.

Neste trabalho, que trata especificamente da fase estática do planejamento, foi implementado um modelo

de programação multi-agente conhecido como Times Assíncronos (*Times-A*) para resolver o problema de inicialização do problema de planejamento da expansão da transmissão. Um *Time-A* consiste de uma rede computacional fortemente cíclica, onde agentes autônomos (aqui, os métodos heurísticos construtivos) cooperam, trabalhando todo o tempo em paralelo, com a partilha permanente de resultados (configurações candidatas) entre os membros do time. Em outras palavras, *Time-A* é um conjunto de processos independentes que trabalham em paralelo todo o tempo, onde as soluções geradas por um processo podem ser reutilizada por outros.

Foram desenvolvidos *Times-A* com variantes dos algoritmos baseados no método de Garver e variantes dos algoritmos de Mínimo Esforço e Mínimo Corte de Carga. Foram feitos testes para três sistemas: Garver (6 barras/15 linhas), Sul brasileiro (46 barras/79 linhas) e Norte-Nordeste brasileiro (89 barras/179 linhas), em uma rede heterogênea de estações SUN, sistema operacional Solaris 2.5.1, com a biblioteca para processamento paralelo PVM 3.3.5. Os resultados foram aplicados aos Algoritmos Genéticos e Busca Tabu, como soluções iniciais factíveis, obtendo-se assim soluções ótimas e quase-ótimas para os sistemas testados. A diversidade e qualidade das soluções obtidas com o uso dos times assíncronos inicializadores, comprovam sua eficácia em gerar soluções iniciais factíveis para os algoritmos combinatoriais.

2.0 TIMES ASSÍNCRONOS

Em (Talukdar e De Souza, 1993) define-se *Time-A* como qualquer super-agente cujos agentes são autônomos, cujas comunicações são assíncronas e cujo fluxo de dados é cíclico. Em outras palavras, *Time-A* é um conjunto de processos independentes que trabalham em paralelo todo o tempo, onde as soluções geradas por um processo podem ser reutilizada por outros. O agente faz sua própria escolha sobre sua seleção de entrada, escalonamento e política de alocação de recursos. Agentes podem ler e escrever informações continuamente nas memórias compartilhadas sem qualquer sincronização entre elas, de maneira que um fluxo contínuo de modificações possa ser realizado tornando possível iterações e “feedback” entre os mesmos.

Pode-se detalhar melhor a idéia contida num *Time-A* com as seguintes considerações (De Souza, 1993):

1. Resultados acumulam nas memórias do fluxo de dados (como nos “blackboards”) para formar populações (como nos algoritmos genéticos).
2. Estas populações são continuamente modificadas por dois tipos de agentes:

- *Agentes de Construção* (que adiciona elementos às populações), e
 - *Agentes de Destruição* (que elimina elementos das populações, mantendo fixo o número de elementos das mesmas).
3. As habilidades em resolver problemas de um fluxo de dados pode ser arbitrariamente repartida entre construção e destruição.
 4. Não existe um controle centralizado, mas novos agentes autônomos pode ser facilmente adicionados ou retirados.
 5. Os fluxos de dados destes *Times-A* não somente produzem boas soluções, mas eles parecem ser “scale-efficient”, isto é, os mesmos podem ser montados para produzirem sempre as melhores soluções pela adição de certos agentes e memórias.

É importante observar que, após a fase de inicialização, cada algoritmo roda em seu próprio ritmo usando os dados disponíveis na(s) memória(s) compartilhada(s). Isto mostra o caráter assíncrono do *Time-A* que pode facilmente ser executado em paralelo; cada agente e memória podem ser alocados nas diferentes máquinas de uma rede de computadores.

3.0 MÉTODOS HEURÍSTICOS CONSTRUTIVOS

3.1 Método de Garver

O modelo formulado em (Garver, 1970), também conhecido como *modelo de transportes*, foi a primeira proposta para planejamento de redes de transmissão que usou programação linear. Esta metodologia consiste basicamente em resolver de maneira aproximada uma versão relaxada do modelo DC. Neste modelo somente se leva em conta a Primeira Lei de Kirchhoff e a capacidade de transmissão das linhas, não sendo levado em conta, portanto, a Segunda Lei de Kirchhoff.

Na metodologia de Garver, todo fluxo que não puder ser transportado pelas ligações normais, fluirão pelas ligações de sobrecarga, pois estas têm capacidades ilimitadas, e só passarão através das ligações de sobrecarga quando for impossível transportá-los pelas ligações normais, já que estas tem custos muito inferiores. Em cada estágio do processo de planejamento, deve-se resolver um problema de programação linear e logo adicionar um circuito na trajetória de maior sobrecarga. O processo é repetido até eliminar todas as sobrecargas.

A vantagem da metodologia de Garver é a simplicidade na implementação do algoritmo pois ela exige somente soluções sucessivas de programação linear. A maior limitação da metodologia é que ela não garante a obtenção da solução ótima do sistema

planejado. Portanto, esta metodologia, em essência, é de natureza heurística.

A metodologia foi usada em três versões (agentes): *GarverA*, *GarverB* e *GarverC*.

Os algoritmos utilizados, são descritos a seguir:

3.1.1 GarverA

1. Faz-se a análise DC (MINOS) do sistema (o banco de dados da rede já considera a presença das barras e linhas fictícias), penalizando-se a configuração pelo custo de um ramo predeterminado.
2. Com os resultados de (1), calcula-se o fluxo para cada ramo do sistema.
3. Ordena-se os ramos pelo valor dos respectivos fluxos, em ordem decrescente.
4. Se o fluxo do ramo mais sobrecarregado for menor que um limite preestabelecido no início do programa, então vá para (6). Caso contrário, o programa oferece duas opções de adição de linhas, dependendo do sistema em questão. Na primeira, pode-se simplesmente adicionar uma linha ao ramo mais sobrecarregado. Na segunda, é feita uma identificação da necessidade da criação de um caminho completo ou somente um ramo, a partir da adição do ramo mais sobrecarregado, a fim de evitar a presença de elementos desconexos no sistema.
5. Voltar para (1).
6. Efetua-se o *refinamento* do sistema, que consiste na verificação da possibilidade de se remover linhas adicionadas desnecessariamente.

3.1.2 GarverB

O algoritmo *GarverB* é semelhante ao *GarverA*, com a diferença que agora a ordenação é realizada baseada no número de linhas requeridas em cada ramo e não no fluxo. Da mesma forma, o critério de parada considera agora uma parcela mínima de linhas permitida.

3.1.3 GarverC

O algoritmo *GarverC* apresenta a mesma estrutura básica que o *GarverB*, exceto pela introdução das seguintes correções:

- correção de caminhos conectando linhas penduradas;
- correção de caminhos entre barras fictícias, para que a capacidade de transmissão das linhas em série seja suficiente;
- adição de linhas cuja fração seja superior a um valor de entrada.

3.2 Método de mínimo esforço

O método do mínimo esforço (Monticelli *et al.*, 1982) baseia-se no fato de que a distribuição dos fluxos em uma rede segue uma *lei de mínimo esforço* que minimiza o produto das reatâncias (*p.u.*) de cada ramo pelo quadrado do respectivo fluxo. Esta função de mínimo esforço, é utilizada aqui como um índice de desempenho para ordenar as adições mais atrativas, da seguinte maneira:

$$Dz_i = y_i^2 Dg_i$$

onde:

y_i é a diferença angular do ramo i antes da adição,

Dg_i é a susceptância do circuito adicionado.

A metodologia foi usada em duas versões (agentes): *minD* e *minE*.

3.2.1 minD

O programa *minD* possui algoritmo semelhante aos anteriores com diferenças, relacionadas ao novo critério, somente nos seguintes itens:

1. Penaliza-se a configuração pela reatância de um ramo predeterminado.
2. Calcula-se o somatório das sobrecargas do sistema e o índice de mínimo esforço para cada ramo do sistema.
3. Ordena-se os ramos pelo valor dos respectivos índices de mínimo esforço.
4. Se o somatório das sobrecargas do sistema for menor que um limite preestabelecido no início do programa, então vá para (6). Caso contrário, escolhe-se aleatoriamente um dos três primeiros ramos da lista ordenada para a adição de uma linha ou de um caminho, com as mesmas considerações anteriores.

3.2.2 minE

A única diferença deste programa em relação ao *minD* consiste na introdução de uma normalização pelo custo de cada ramo, quando do cálculo do índice de mínimo esforço.

3.3 Método de mínimo corte de carga

De maneira semelhante ao Método de Mínimo Esforço, este método (Pereira *et al.*, 1985) realiza a adição de linhas selecionadas de acordo com um índice de sensibilidade que permite encontrar linhas mais atrativas. O índice utilizado é:

$$IS_{mcc} = (p_i - p_j)(q_i - q_j)$$

onde:

p_j é o multiplicador de Lagrange da j -ésima restrição

$$(Bq + g + r = d),$$

q_j são os ângulos de tensão de barra obtidos na análise DC do sistema.

Neste método fez-se uma única versão (agente): *mccF*.

3.3.1 mccF

O programa *mccF* também mantém a mesma estrutura dos anteriores com diferenças, associadas ao novo índice, somente nos itens:

1. Penaliza-se a configuração pela reatância de um ramo predeterminado.
2. Calcula-se o corte de carga do sistema (via MINOS), e o índice de mínimo corte de carga, aqui já normalizado pelo custo, para cada ramo do sistema.
3. Ordena-se os ramos pelo valor dos respectivos índices de corte de carga.
4. Se o corte de carga do sistema for menor que um limite preestabelecido no início do programa, então vá para (6). Caso contrário, escolhe-se aleatoriamente um dos três primeiros ramos da lista ordenada para a adição de uma linha ou de um caminho, com as mesmas considerações anteriores.

4.0 MODELO DE TIME-A PROPOSTO

A estrutura do *Time-A* implementada é composta pelos seguintes objetos:

- *memória central* - memória compartilhada pelos agentes, a qual é formada por uma população de configurações. O tamanho desta população é constante e determinado previamente pelo usuário.
- *agentes de construção* - são os diversos algoritmos heurísticos construtivos implementados (*GarverA*, *GarverB*, etc.).
- *agente destruidor* - mantém constante o tamanho da população.

Um exemplo de *Time-A*, dentro da estrutura elaborada, é mostrado na Figura 1. Os agentes são representados por setas que indicam um fluxo cíclico em relação à memória central, ou seja, os mesmos lêem os dados da memória, alteram esses dados, e os reescrevem na mesma. A memória central, representada por um retângulo, contém a população de configurações. Os algoritmos implementados na elaboração do time assíncrono são representados por linhas tracejadas, definindo processos distintos. Daí, observa-se que o

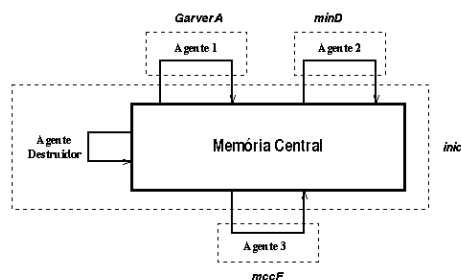


FIGURA 1 - Exemplo de *Time-A* com três agentes.

agente destruidor e a memória central compõem o processo *inic*. Note que o agente destruidor não é um processo independente como os outros agentes, mas sim uma das funções do processo *inic* que age para manter constante o tamanho da população.

A implementação paralela usa o modelo *Mestre/Escravo*, pois o *inic* dá início ao processamento do time, partindo um grupo de agentes predeterminados pelo usuário. Porém, é possível a adição/remoção dos agentes de forma manual, o que reforça a característica de autonomia dos mesmos (agentes entram ou saem do time sem interferir no funcionamento uns dos outros). A remoção manual dos agentes é feita por um programa auxiliar, enquanto que a adição dos mesmos é feita executando-os diretamente no “prompt” do sistema operacional.

O programa *inic* simula uma memória compartilhada e utiliza um controle de acesso aos dados para evitar conflitos ao se tentar acessá-la. Dessa forma, dois agentes não podem ler ou escrever ao mesmo tempo. Sendo assim, quando um agente estiver acessando a memória central, os outros necessariamente esperam numa fila do tipo FIFO (“First-In-First-Out”). Outra característica importante é o fato de que, a cada instante do processamento do *Time-A*, cada configuração (elemento da população) é atualizada somente por um agente, isto é, dois agentes não podem trabalhar sobre os mesmos dados simultaneamente. Dessa forma, o número de membros do time nunca deve exceder o número de configurações da população. Se assim o for, o *inic* automaticamente se encarrega de eliminar gradativamente os agentes excedentes. A Figura 2 dá uma idéia geral do funcionamento do processo *inic*.

Quanto aos agentes de construção, de uma maneira geral, servem para efetuar os cálculos sobre os dados da memória central, segundo uma heurística própria, atualizando-os até se satisfazer um determinado critério de parada. A Figura 3 apresenta a estrutura de funcionamento dos agentes em geral. Eles trabalham sinergicamente para obter configurações iniciais diversificadas em menor tempo, não visando necessa-

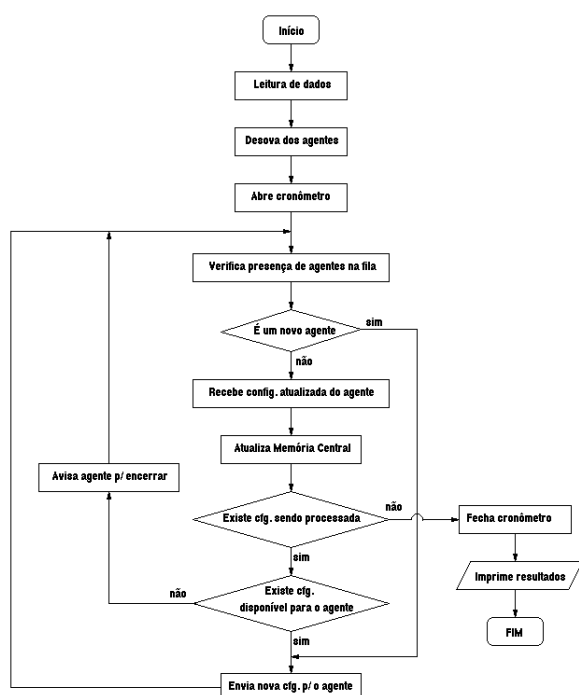
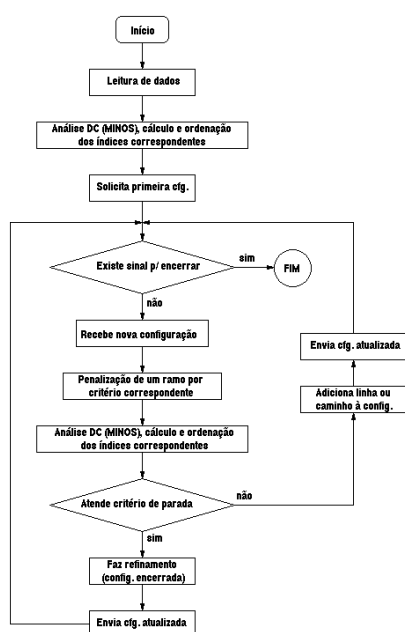
FIGURA 2 - Estrutura básica do programa *inic*.

FIGURA 3 - Estrutura básica dos agentes de construção.

riamente a obtenção de um ótimo global, função esta do problema do planejamento como um todo. Os agentes podem ser combinados em quantidades diferentes entre as máquinas da rede.

5.0 TESTES E RESULTADOS

Foram realizadas diversas simulações para sistemas de pequeno, médio e grande porte, respectivamente: Garver (6 Barras – 15 Linhas), Sul brasileiro (46

Barras – 79 Linhas) e Norte-Nordeste brasileiro (89 Barras – 179 Linhas). Tabelas, gráficos e configurações ótimas referentes as simulações estão disponíveis em (De Oliveira *et al.*, 1998).

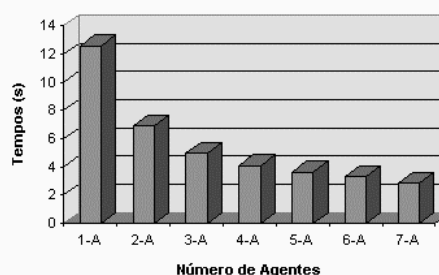
Para mostrar a característica “scale-efficient” do time, rerepresenta-se aqui a Tabela 1 e a Figura 4, referentes ao agente *GarverA*.

TABELA 1 – Tempos para diferentes números do *garverA*

Simulação	Sistema Garver (6 Barras – 15 Linhas)						
	Combinações (Tempos em segundos)						
	1-A	2-A	3-A	4-A	5-A	6-A	7-A
1	12,54	7,33	5,58	4,57	4,00	3,53	2,70
2	12,61	6,80	4,93	4,12	3,53	3,15	2,93
3	12,62	6,72	4,94	4,05	3,81	3,22	3,01
4	12,60	6,87	4,98	4,28	3,64	3,22	2,90
5	12,62	7,01	4,85	3,96	3,45	3,38	2,76
6	12,63	6,70	4,97	3,95	3,60	3,38	2,75
7	12,56	6,71	4,73	4,03	3,56	3,22	2,88
8	12,62	6,93	4,74	3,98	3,74	3,29	2,83
9	12,58	7,00	4,96	4,07	3,43	3,55	3,02
10	12,55	7,07	4,83	3,91	3,38	3,13	2,83
Méd.	12,59	6,91	4,95	4,09	3,61	3,31	2,86

NOTAS: População: 8 configurações

Obteve-se a configuração ótima em todos os casos.

FIGURA 4 – Tempos para diferentes números do *garverA*

Neste trabalho, foram realizadas algumas simulações com os algoritmos Busca Tabu e Algoritmos Genéticos (Gallego, 1997) para verificar a eficiência no uso dos *Times-A* como um algoritmo inicializador. Ou seja, foram geradas configurações iniciais factíveis através de um *Time-A* composto com todas as variantes dos métodos heurísticos construtivos descritos anteriormente. Em seguida, utilizou-se estas configurações para inicializar os algoritmos citados, comparando os resultados com aqueles obtidos, sem o uso do time inicializador.

As Figuras 5 e 6, apresentam alguns dos resultados obtidos para os sistemas Garver e Sul brasileiro, simulados sem e com o auxílio do *Time-A*, para os algoritmos Busca Tabu e Algoritmos Genéticos, respectivamente; enquanto que a Figura 7 apresenta os resultados obtidos para o sistema Norte-Nordeste brasileiro, para os Algoritmos Genéticos, com 60 configurações iniciais geradas previamente.

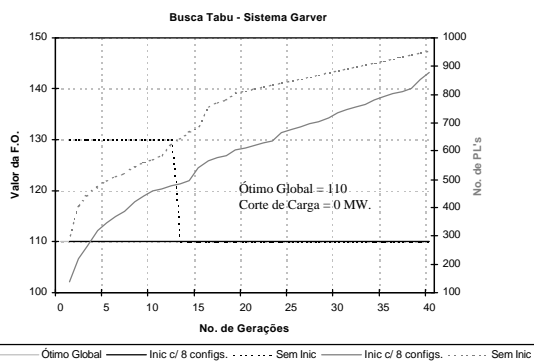


FIGURA 5—Resultado sem e com *Inic* para Garver/BT

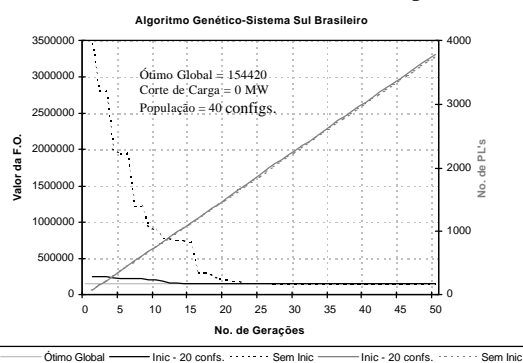


FIGURA 6 – Resultado sem e com *Inic* para Sul/AG Veja, nas figuras anteriores, que o uso do *Time-A*, possibilitou a obtenção do ótimos globais, para ambos sistemas, em um número menor de ciclos ou gerações (ou seja, com menos resoluções de PL's).

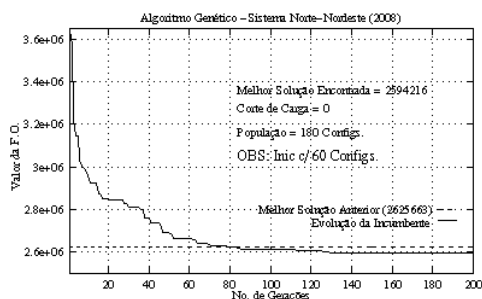


FIGURA 7 – Resultado com *Inic* para Norte-Nord/AG

Na Figura acima, selecionou-se 60 melhores configurações de diferentes simulações prévias do *Time-A*, como soluções iniciais factíveis para o sistema Norte-Nordeste brasileiro, simulado aqui com os Algoritmos Genéticos. Verifica-se que com o auxílio destas configurações iniciais, obteve-se um resultado melhor que o apresentado em (Gallego, 1997).

6.0 CONCLUSÕES

Neste trabalho, aplicou-se times assíncronos aos algoritmos heurísticos construtivos (Garver, Mínimo Esforço e Mínimo Corte de Carga) de planejamento da expansão da transmissão, para a implementação de programa inicializador. Este, por sua vez, é indispensável para o uso de algoritmos combinatoriais

como Busca Tabu e Algoritmos Genéticos, quando se trabalha com sistemas de grande porte, pois a eficácia da busca no espaço de soluções é função das soluções iniciais factíveis.

Dos testes realizados, concluiu-se que é possível implementar o modelo de times assíncronos a fim de se obter um desempenho melhor que o apresentado por algoritmos de maneira isolada. Os testes comprovam também a eficácia no uso das configurações resultantes dos *Times-A* nos algoritmos combinatoriais tais como Algoritmos Genéticos e Busca Tabu.

Com a tendência cada vez maior do uso de redes de computadores nas concessionárias de energia elétrica, aliado a disponibilidade de bibliotecas para processamento paralelo, tem-se o ambiente propício para a utilização de modelos de programação distribuída, como times assíncronos, nos problemas complexos e de natureza combinatoria como o problema de planejamento da expansão da transmissão de energia elétrica.

7.0 REFERÊNCIAS BIBLIOGRÁFICAS

TALUKDAR, S.N. e DE SOUZA, P.S. (1993). Scale efficient organizations, *Procs. of the Int. Conference on Systems, Man and Cybernetics*, New Y.:IEEE, pp.1458-1463.

DE SOUZA, P.S. (1993). *Asynchronous organization for multi-algorithm problems*, Pittsburgh, Department of Electrical and Computer Engineering, Carnegie Mellon University. (Dissertation, Ph.D.)

GARVER, L.L. (1970). Transmission network estimation using linear programming, *IEEE Trans. on PAS*, v.PAS-89, pp. 1688-1697.

MONTICELLI, A. *et al.* (1982). Interactive transmission network planning using a least-effort criterion, *IEEE Trans.on PAS*, v.PAS-101, n.10, pp. 3919-3925.

PEREIRA, M.V.F. e PINTO, L.M.V.G. (1985). Application of sensibility analysis of load supplying capability to interactive transmission expansion planning, *IEEE GALLEGO, R.A. (1997). Planejamento a longo prazo de sistemas de transmissão usando técnicas de otimização combinatorial*, Campinas: Faculdade de Eng. Elétrica e de Computação da UNICAMP, 243p. (Tese, Dr. Eng. Elétrica)

DE OLIVEIRA, S.A. *et al.* (1998). Times assíncronos aplicados a métodos heurísticos construtivos de planejamento da expansão da transmissão.

(SEPARATA)

DADOS BIOGRÁFICOS

Sérgio Azevedo de Oliveira – Nascido em Murutinga do Sul-SP, em 1958, Engenheiro Eletricista pela UNESP/Ilha Solteira em 1981. Mestre em Eng. Elétrica pela UFSC em 1989. Atualmente fazendo doutorado na UNICAMP. Professor junto ao Depto. de Engenharia Elétrica, da FEIS/UNESP, desde 1982. *Student Member* do IEEE, Membro da SBA. Áreas de atuação: Planejamento da Expansão da Transmissão, Processamento Paralelo e Distribuído, Meta-Heurísticas, Times Assíncronos.