

PARALELIZAÇÃO DE UM ALGORITMO DE CALIBRAÇÃO MULTI-OBJETIVO PARA A PREVISÃO OPERACIONAL DE VAZÃO DE RIOS

Aurélio W. T. de Noronha^{1} & Alexandre C. Costa² & Eduardo S. P. R. Martins³ & Fco. Venicius F. Barros⁴ & José M. Rodrigues⁵*

Resumo – Neste trabalho, o MOPSO, um algoritmo evolucionário para a calibração multi-objetivo de modelos hidrológicos, foi migrado para a linguagem FORTRAN e executado com o WASA-SED, um modelo hidrológico semi-distribuído, em uma plataforma UNIX. O algoritmo MOPSO foi reescrito com suporte para ser executado com vários processadores (paralelização). Para a avaliação do ganho computacional da paralelização do sistema MOPSO/WASA-SED, dois experimentos numéricos foram realizados para a bacia hidrográfica do Coreaú no Estado do Ceará, em que o número de processadores foram diferentes (um processador e oito processadores). O tempo de processamento foi menor cerca de 80% quando o experimento foi executado com oito processadores, em relação ao tempo com um processador. Com esse resultado, a aplicação do conjunto MOPSO/WASA-SED pode ser ampliada para todas as bacias do Estado do Ceará com um tempo computacional relativamente pequeno, permitindo também que previsões de vazões sejam realizadas em tempo hábil para a tomada de decisão em gerenciamento de recursos hídricos no Estado do Ceará.

Palavras-Chave – MOPSO, WASA-SED, Paralelização

PARALLELIZATION OF AN ALGORITHM OF CALIBRATION MULTI-OBJECTIVE FOR STREAMFLOW OPERATIONAL FORECASTING

Abstract – In this work, the MOPSO, an algorithm for multi-objective calibration of hydrological models, was written to FORTRAN language and run with the WASA-SED, a streamflow forecasting model, at an UNIX platform. The MOPSO algorithm was modified to be able to run with several cores (parallel computer) and to couple with the WASA-SED model. In order to evaluate the processing time of the system MOPSO/WASA-SED, we applied it to the Coreaú Watershed in the State of Ceará considering a) one core and b) eight cores. We found that both the numerical experiments a) and b) converged to the same model performance and calibrated parameters, but the processing time with eight cores was about 1/5 of that with one core. Therefore, the MOPSO/WASA-SED system can be applied to all watersheds in Ceará with a relatively low processing time. This is also useful to forecast streamflow out in time for decision making in water resources management.

Keywords – MOPSO, WASA-SED, Parallelization

¹Fundação Cearense de Meteorologia e Recursos Hídricos – FUNCEME, awildson@gmail.com

²Fundação Cearense de Meteorologia e Recursos Hídricos – FUNCEME, cunhacos@gmail.com

³Fundação Cearense de Meteorologia e Recursos Hídricos – FUNCEME, espr.martins@gmail.com

⁴Universidade Federal do Ceará – UFC, Departamento de Engenharia Hidráulica e Ambiental - DEHA, veniciusfb@gmail.com

⁵Fundação Cearense de Meteorologia e Recursos Hídricos – FUNCEME, marcelorodriguess@gmail.com

1. INTRODUÇÃO E OBJETIVOS

Em regiões semi-áridas, esforços têm sido feitos no sentido de aperfeiçoar a previsão de vazão de bacias hidrográficas de médio e grande porte para o gerenciamento de água e alocação dos recursos hídricos, principalmente, para ajudar a tomada de decisões em períodos com grandes demanda de água.

As tecnologias de processamento de dados e de *hardware* evoluem constantemente. O tempo para realizar cálculos e armazenamento diminui, e a capacidade de realizar essas tarefas aumenta. A partir dessas vantagens tecnológicas, a informação científica aumenta potencialmente e os aperfeiçoamentos nos algoritmos se tornam cada vez mais realísticos, seja no tratamento físico das soluções ou num tempo menor para o processamento de informações.

Um desses aperfeiçoamentos é o uso de vários processadores para reduzir o tempo de processamento de simulações numéricas para previsões. O MOPSO, neste trabalho, foi migrado para a linguagem FORTRAN e executado numa plataforma UNIX. O código foi escrito de forma que o MOPSO juntamente com o WASA-SED fosse executado por vários processadores, ou seja, executado de forma paralela.

O objetivo geral é reescrever o código fonte do MOPSO (originalmente em MATLAB[®]) para linguagem FORTRAN[®] com suporte para o paralelismo (N processadores). E o objetivo específico é avaliar o ganho de tempo computacional executando dois experimentos numéricos: um com um processador e outro com oito processadores.

2. O MOPSO (*Multi Objective Particle Swarm Optimization*)

Inicialmente, um conjunto de soluções (população de indivíduos) é gerado aleatoriamente dentro do espaço de busca do problema e avaliado segundo a função objetivo em questão, sendo esse valor utilizado como aptidão da solução (indivíduo)[Barros *et al* (2010)].

Em seguida é feita uma busca dentre essas soluções para identificar qual possui o menor valor de aptidão, visto que se procura minimizar a função objetivo. Sendo assim, aquela que possui o menor valor de aptidão é a melhor solução (indivíduo), sendo esta tida como a melhor solução global (guia do bando).

A concepção do algoritmo introduz o conceito de melhor individual de cada solução (partícula), que é a melhor posição já ocupada até o presente momento na evolução da busca. Logo, na primeira evolução da população inicial, ou seja, no primeiro passo, o melhor individual de cada solução é seu próprio valor inicial.

As evoluções das soluções (partículas) obedecem à seguinte expressão[Alvarez *et al.* (2005)]

$$x_n^{(t+1)} = x_n^{(t)} + \chi v_n^{(t)} + \varepsilon_n^{(t)} \quad (1)$$

em que $x_n^{(t)}$ e $x_n^{(t+1)}$ são as n -ésimas soluções (posições da partícula) nas iterações t e $t + 1$, respectivamente, $v_n^{(t)}$ é a velocidade da partícula n , χ é o fator que determina a magnitude do passo (valor entre 0 e 1), e $\varepsilon_n^{(t)}$ representa uma pequena perturbação estocástica, conhecida como “fator de turbulência”, o qual ajuda a solução (partícula) a sair de ótimos locais e contribuir na diversidade da busca.

O passo da solução (velocidade da partícula) citado acima é determinado a cada interação, pela seguinte expressão:

$$v_n^{(t+1)} = wv_n^{(t)} + c_1r_1(P_n - x_n^t) + c_2r_2(G_n - x_n^t) \quad (2)$$

em que $v_n^{(t)}$ e $v_n^{(t+1)}$ são os passos da solução (velocidades da partícula) nas iterações t e $t + 1$, respectivamente, w é a inércia da solução (partícula), c_1 e c_2 são constantes que controlam a influência do passo (velocidade) individual e global, e r_1 e r_2 são números aleatórios uniformemente distribuídos entre 0 e 1.

Após a evolução das soluções (partículas), é feita a avaliação da função objetivo das mesmas e atualizado o valor de suas respectivas aptidões. Dentre as novas soluções é selecionada aquela que tem o menor valor de aptidão, ou seja, a melhor solução (guia do bando) do atual conjunto de soluções (bando).

É realizado um teste comparativo para saber se a melhor solução gerada é melhor que a melhor solução atual. Caso seja “verdade”, a melhor solução global é atualizada, caso contrário, permanece com a solução atual. As melhores posições individuais de cada solução são também comparadas e atualizadas seguindo a mesma lógica.

Com o intuito de abranger problemas multiobjetivo, algumas modificações no algoritmo *Particle Swarm Optimization* (PSO) foram propostas por [Alvarez *et al.* (2005)]. Essas modificações são tratadas em maiores detalhes a seguir.

Inicialmente, como no PSO é gerada uma população inicial de soluções (bando) aleatoriamente, e, então, avaliada segundo as diversas funções objetivo em questão.

A partir dos conceitos de dominância de Pareto, determinam-se quais soluções são não-dominadas na população inicial. Estas soluções são armazenadas em um repositório de soluções não-dominadas (frente de Pareto).

Dentro dessa nova abordagem, cada solução (partícula) possui um melhor global, determinado pela seguinte regra:

- Se a solução (partícula) é dominada por soluções da frente, o melhor global para essa solução (partícula) é determinado aleatoriamente entre as soluções da frente que a dominam;
- Se a solução (partícula) entra na frente, o melhor global para ela é uma solução da frente determinada aleatoriamente.

O melhor individual de cada solução (partícula) é determinado, seguindo as seguintes regras:

- Se a nova posição da solução (partícula) domina a melhor posição individual, a melhor posição individual é alterada para a nova posição;
- Se a nova posição da solução (partícula) não domina e nem é dominada pela melhor posição individual, então a melhor posição individual é alterada para a nova posição;
- Se a nova posição é dominada pela melhor posição individual, esta permanece a mesma.

O processo de evolução das soluções (partículas) na busca se dá da mesma forma que no PSO. No primeiro passo no processo de busca, o melhor individual de cada solução é sua posição inicial e o melhor global segue a regra anteriormente descrita. Similarmente à abordagem uniobjetivo, o processo é repetido até que o critério de parada, previamente determinado, seja satisfeito.

3. O MODELO WASA-SED

O Modelo WASA foi inicialmente desenvolvido para simular a disponibilidade de água em regiões semi-áridas com o objetivo de estudar o balanço hídrico de longo prazo em bacias hidrográficas de grande porte. O modelo tem um *framework* dinâmico que inclui uma hierarquia conceitual da modelagem hidrológica que captura a variabilidade de uso da terra e identifica os processos hidrológicos laterais e verticais dominantes [Güntner e Bronstert (2004)]. Em versões recentes, a erosão, o transporte de sedimentos e os processos de retenção foram incluídos no WASA, sendo chamado de modelo WASA-SED [Mueller *et al.* (2010)]. Uma descrição detalhada do modelo WASA-SED pode ser encontrada nas citações mencionadas.

4. ACOPLAMENTO DO WASA-SED E MOPSO

O MOPSO em sua primeira versão foi escrito na linguagem MATLAB[®]. Inicialmente, o código fonte é estruturado e escrito para ser executado com apenas um núcleo na plataforma WINDOWS[®]. O MOPSO é um algoritmo de determinação de melhores soluções entre partículas e funções.

As funções calculadas (coeficiente Nash-Sutcliffe de vazões mensais em diferentes estações fluviométricas) são determinadas com base nos valores de vazão dos afluentes simulados do WASA-SED. O MOPSO interage bidirecionalmente com o WASA-SED, onde as partículas selecionadas, através do avanço da frente de Pareto, são fornecidas ao WASA. O objetivo do MOPSO está em convergir para soluções melhores das funções objetivas.

O WASA-SED, escrito em linguagem FORTRAN[®], é compilado para sua finalidade – simulação de vazão das bacias – não necessitando de bibliotecas de acesso para sua execução. O WASA requer um tempo de simulação que é proporcional ao número de bacias e do passo de tempo escolhido no experimento (caso diário).

Com os valores iniciais do conjunto de partículas, o WASA-SED é executado para cada partícula. O MOPSO lê os resultados gerados pelo WASA-SED em uma iteração e calcula as funções objetivo daquela iteração para cada partícula. Dada a primeira matriz de funções objetivo, então, inicia-se o avanço das partículas selecionando as melhores soluções através da frente de Pareto (Figura 1).

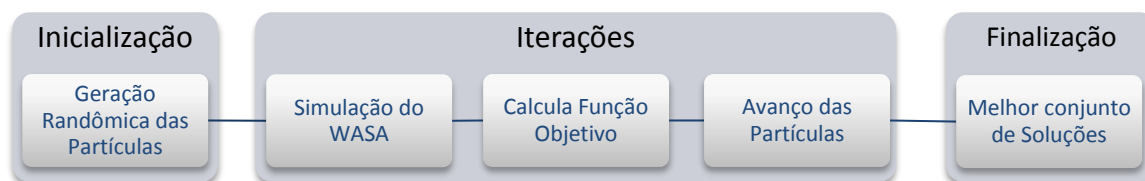


Figura 1 – Fluxograma de processos no MOPSO com o WASA-SED.

O MOPSO encerra-se quando alcança sua última iteração. E finalmente, armazena-se o conjunto de partículas e funções objetivos que convergem para melhor solução.

5. SOLUÇÃO COM A PARALELIZAÇÃO

O WASA-SED possui um tempo particular para realizar uma simulação. Em cada partícula, o WASA-SED é acionado e, portanto, o tempo acumulado de processamento, até aqui, é proporcional ao número de partículas multiplicado pelo tempo particular de cada simulação. E por fim, acrescente ao tempo acumulado o número de iterações do MOPSO, até que, este termine a simulação.

O tempo acumulado de processamento será demasiadamente grande caso o MOPSO seja executado apenas com um núcleo de um computador (Figura 2). A computação paralela é uma técnica poderosa para reduzir esse tempo de processamento na execução do MOSPO.

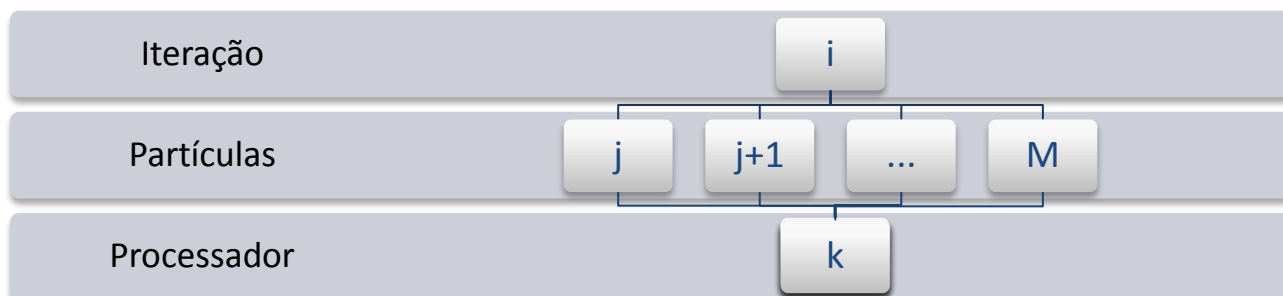


Figura 2 – Fluxograma de solução sequencial para a execução do WASA-SED em cada partícula. Neste, o tempo é maior, pois os processos são resolvidos apenas por um processador k.

A computação paralela é uma forma de computação em que vários cálculos são feitos simultaneamente[Fabbri (2011)]. Dentre os tipos possíveis de paralelismo, a paralelização de tarefas foi utilizada no algoritmo do MOPSO. Para que um processo, num algoritmo, seja paralelizado é necessário que seu código fonte seja escrito de maneira adequada para esse fim (Figura 3).

No MOPSO, a frente de Pareto não é o processo que demanda maior tempo de processamento, mas a execução do WASA-SED para cada partícula. O laço de execução do WASA-SED em cada partícula não requer uma dependência entre uma partícula ou outra. Neste sentido, o MOPSO foi reescrito, de modo que, o WASA-SED pudesse ser executado simultaneamente em cada partícula.

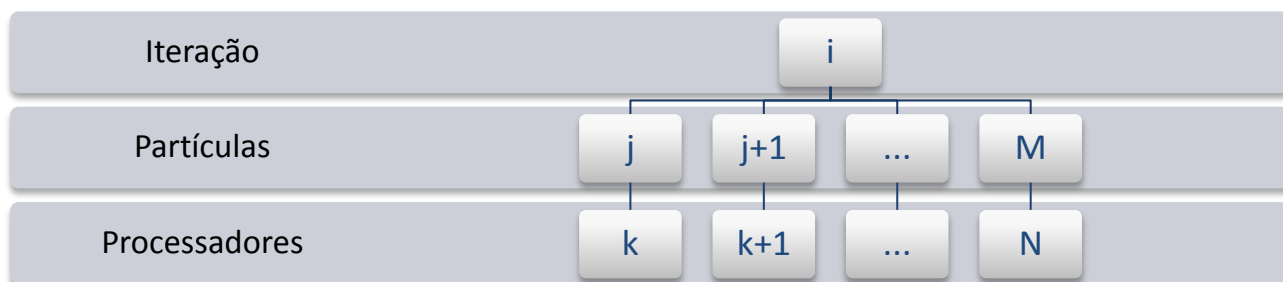


Figura 3 – Fluxograma de solução paralela para a execução do WASA-SED em cada partícula. Neste esquema, o tempo de processamento é menor, pois os processos são distribuídos a N processadores.

6. AVALIAÇÃO DA PARALELIZAÇÃO

O código fonte do MOPSO foi transcrito para a linguagem FORTRAN[®] para ser executado no ambiente UNIX. Após a transcrição, o código fonte foi adaptado para a paralelização. No MOPSO o laço de chamada do WASA-sed para cada partícula foi reescrito de modo que as chamadas do WASA-SED fossem distribuídas para N processadores.

Dois experimentos numéricos de calibração foram realizados para a bacia hidrográfica do Coreaú, Estado do Ceará, Brasil (Figura 4); avaliando a solução com a paralelização. Primeiro, uma simulação com apenas um processador e segundo, uma simulação de mesma configuração com oito processadores.



Figura 4 – Localização da Bacia Hidrográfica do Coreaú no Estado do Ceará.

Os parâmetros do WASA-SED e do MOPSO são apresentados nas Tabelas 1 e 2.

Tabela 1 – Parâmetros utilizados nas simulações do WASA-SED.

Parâmetro	Valor
Ano Inicial	1980
Ano Final	2010

Número de sub-bacias	18
Passo de Tempo	Diário

Tabela 2 – Parâmetros utilizados no experimento do MOPSO.

Parâmetro	Valor
Número de Partículas	100
Dimensão das Partículas	14
Número de Iterações	25
Número de Estações Fluviométricas	3
Número de Funções Objetivo	3

Com a migração de linguagem e plataforma, ou seja, migrando o código fonte do MATLAB executado no WINDOWS para FORTRAN executado no UNIX, um ganho de tempo computacional considerável foi obtido. O experimento numérico para a bacia do Coreaú executado no UNIX é cerca de 35% (tabela não mostrada) menor que o tempo na versão executada na plataforma anterior à migração (com o uso de um processador em cada plataforma).

Já o uso de oito processadores com as mesmas configurações no experimento numérico foi cerca de 80% menor quando comparado ao experimento com apenas um processador (Tabela 3). Note que os dois experimentos numéricos convergiram para os mesmos valores de partículas e funções objetivo.

Tabela 3 – Tempo de processamento nos experimentos numéricos do MOPSO.

Experimento	Processadores	Hora Inicial	Hora Final	Tempo
Coreau – 8A	8	16:43:00	18:04:00	01:21:00
Coreau – 1A	1	14:30:00	21:00:00	06:30:00

Os resultados mostram que o uso do algoritmo MOPSO usado em paralelo com oito processadores teve um tempo reduzido considerável para a calibração da bacia de Coreaú com o modelo WASA-SED. E com um cluster de computadores este tempo pode ser ainda menor.

Com esse resultado, a aplicação do conjunto MOPSO/WASA-SED pode ser estendido para todas as bacias do Estado do Ceará com um tempo computacional relativamente pequeno, permitindo também que as previsões de vazões sejam realizadas em tempo hábil para a tomada de decisão em gerenciamento de recursos hídricos no Estado do Ceará.

REFERÊNCIAS

ALVAREZ-BENITEZ, J. E., EVERSON, R. M., & FIELDSEND, J. E. (2005). A MOPSO algorithm based exclusively on paretodominance concepts. *In* EMO (pp 459–473). Apresentadona Third International Conference on Evolutionary Multi-Criterion Optimization, Guanajuato, México: LNCS 3410, Springer-Verlag.

Barros, F., Martins, E., Nascimento, L., & Reis, DirceuSilveira, J. (2010). Use of Multiobjective Evolutionary Algorithms in Water Resources Engineering. Org. por N. Nedjah, L. Santos Coelho, & L. MacedoMourelle, *Multi-Objective Swarm Intelligent Systems* (Vol 261, pp 45–82). Springer

Berlin Heidelberg. Obtido de http://dx.doi.org/10.1007/978-3-642-05165-4_3

Fabbri, R. (2011). Introduction to Parallel Computing. *Introduction to Parallel Computing*. Wiki. Obtido 7 de Maio de 2013, de http://wiki.nosdigitais.teia.org.br/Introduction_to_Parallel_Computing

Güntner, A., & Bronstert, A. (2004). Representation of Landscape Variability and Lateral Redistribution Processes for Large-Scale Hydrological Modelling in Semi-Arid Areas. *Journal of Hydrology*, 136–161.

Mueller, E. N., Güntner, A., Francke, T., & Mamede, G. (2010). Modelling sediment export, retention and reservoir sedimentation in drylands with the WASA-SED model. *Geoscientific Model Development*, 3(1), 275–291. doi:10.5194/gmd-3-275-2010