



**زیربرنامه:**

TopologicalCleanUp

|  |  |  |
| --- | --- | --- |
| **توسعه دهندگان** | کورش مرادیان | C:\Users\Kourosh\Desktop\63.png |
| **تهیه کنندگان مستند** | کورش مرادیان | |
| **تاییدکنندگان** | مرتضی نامور | |
| **تاریخ تنظیم سند** | 22/09/1396 | |
| **شناسه سند** |  | |
| **زبان برنامه‌نویسی** | **Fortran 90/95** | |

1. وظایف

در این زیربرنامه مجموعه‏ای از الگوریتم‏ها برای یافتن الگوهای ساختاری نامناسب در شبکه و اصلاح آنها در جهت بهبود توپولوژی شبکه به کار گرفته می‏شوند.

1. توضیحات و تئوری­ها

یکی از ویژگی‏های منفی الگوریتم‏های غیرمستقیم برای ایجاد شبکه‏های چهارضلعی این است که معمولاً این روش‏ها نودهای نامعمول زیادی به شبکه اضافه می‏کنند. نودهای نامعمول در شبکه‏های چهارضلعی به نودهایی اتلاق می‏شوند که درجه آنها کمتر یا بیشتر از چهار است. هرچند که به واسطه تغییر اندازه المان‏های مجاور در شبکه وجود چنین نودهایی ضروری است اما زیاد بودن آنها بر کیفیت شبکه تاثیر منفی دارد. الگوریتم Q-Morph هم به عنوان یک روش غیر مستقیم از مجموعه‏ای از الگوریتم‏های ارائه شده در کارهای پیشین برای شناسایی و ترمیم الگوهای ساختاری نامناسب ایجاد شده در شبکه استفاده می‏کند. این عملیات‏ها به سه دسته عملیات‏های اساسی، عملیات‏های چند گامی و عملیات‏های مرزی دسته‏بندی می‏شوند.

مسئله‏ای که در منابع به آن پرداخته نشده است ترتیب اجرای آنها و همچنین تعداد تکرارهای مورد نیاز برای حصول اطمینان از بررسی شدن تمامی الگوهاست زیرا حذف یک الگو ممکن است الگوی دیگری را ایجاد کند. برای حل این مشکلات ما از شبه کد پیشنهادی در ‏جدول (1) استفاده می‏کنیم.

با استفاده از این روند پیشنهادی نیازی به تعیین کردن تعداد تکرارها نخواهد بود و همچنین استفاده از عملیات‏ها با اولویت قرار دادن عملیات‏های اساسی بر عملیات‏های چندگامی و در نهایت عملیات‏های مرزی در بررسی کردن بخش‏های شبکه روند مناسبی خواهد بود.

1. شبه کد پیشنهادی برای اجرای الگوریتم‏های پاکسازی توپولوژیکی

|  |
| --- |
| **Algorithm: Topological Cleanup**  **Input:** Quadrilateral Mesh (Me)  **Output:** Quadrilateral Mesh (Me)  1: begin  2: do while no more patterns found  3: do  4: if (basic pattern found) then  5: remove pattern  6: exit loop  7: else if (multiple operation pattern found) then  8: remove pattern  9: exit loop  10: else if (boundary pattern found) then  11: remove pattern  12: exit  13: end if  14: end do  15: end while  16: end |

1. بخش­های زیربرنامه

در این قسمت تمام بخش های زیربرنامه مطابق با شماره گذاری موجود در برنامه کامپیوتری ارائه شده است. برای آشنایی کامل با الگوهای مورد نظر در ادامه و نحوه ترمیم هر کدام به جزئیات زیربرنامه‏های مرتبط با آنها رجوع کنید.

1. حذف نودهای درجه دو

در این بخش با بررسی المان‏های شبکه نودهایی که فقط به دو لبه متصل هستند یا درجه دو هستند با فراخوانی زیربرنامه NodeElimination حذف می‏شوند.‏

1. حذف المان

در این بخش المان‏هایی که دو نود غیر مجاور آنها درجه سه باشند با فراخوانی زیربرنامه ElementElimination حذف می‏شوند.

1. حذف لبه‏های اضافی

در این بخش المان‏هایی که دو نود مجاور آنها درجه سه باشند شناسایی می‏شوند و لبه مورد نظر آنها با فراخوانی زیربرنامه SegmentElimination حذف و اصلاحاتی در شبکه ایجاد می‏شود.

1. اصلاح الگوهای دو نود درجه سه متصل به یک نود درجه چهار

در این بخش الگوهایی که در آنها دو نود درجه سه مشترکاً به یک نود درجه چهار متصل هستند شناسایی می‏شوند و با فراخوانی زیربرنامه ThreeEdgedNodesDividedByFourEdgedNode حذف و اصلاحاتی در شبکه ایجاد می‏شود.

1. اصلاح الگوهای 3-5-oppo-3

در این بخش الگوهای 3-5-oppo-3شناسایی می‏شوند و با فراخوانی زیربرنامه ThreeFiveOppoThree حذف و اصلاحاتی در شبکه ایجاد می‏شود.

1. اصلاح الگوهای 5-3-oppo-5

در این بخش الگوهای 5-3-oppo-5 شناسایی می‏شوند و با فراخوانی زیربرنامه FiveThreeOppoFive حذف و اصلاحاتی در شبکه ایجاد می‏شود.

1. اصلاح الگوی TwoCollapses

در این بخش الگوهایی که با انجام دو عملیات Collapse قابل بهبود هستند شناسایی و با فراخوانی زیربرنامه TwoCollapsesOperation اصلاح می‏شوند.

1. اصلاح الگوی TwoFiveOppoThree

در این بخش الگوهای مورد نظر شناسایی و با فراخوانی زیربرنامه TwoFiveOppoThree اصلاح می‏شوند.

1. اصلاح الگوی ThreeFiveOppoFiveThree

در این بخش الگوهای مورد نظر شناسایی و با فراخوانی زیربرنامه ThreeFiveOppoFiveThree اصلاح می‏شوند.

1. اصلاح الگوی ThreeFiveFive

در این بخش الگوهای مورد نظر شناسایی و با فراخوانی زیربرنامه ThreeFiveFive اصلاح می‏شوند.